

**RL-TR-96-123**  
**Final Technical Report**  
**July 1996**



# **MULTISENSOR TRACK-TO-TRACK FUSION FOR AIRBORNE SURVEILLANCE SYSTEMS**

**Nova Research Corporation**

**R.K. Saha (Nova Research Corporation)**  
**K.C. Chang (George Mason University)**  
**M.M. Kokar (Northeastern University)**

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

**19960924 061**

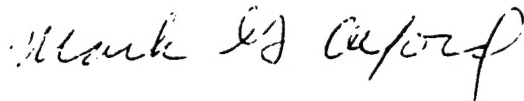
**DTIC QUALITY INSPECTED 2**

**Rome Laboratory**  
**Air Force Materiel Command**  
**Rome, New York**

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be releasable to the general public, including foreign nations.

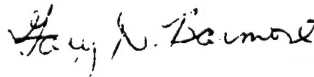
RL-TR- 96-123 has been reviewed and is approved for publication.

APPROVED:



MARK G. ALFORD  
Project Engineer

FOR THE COMMANDER:

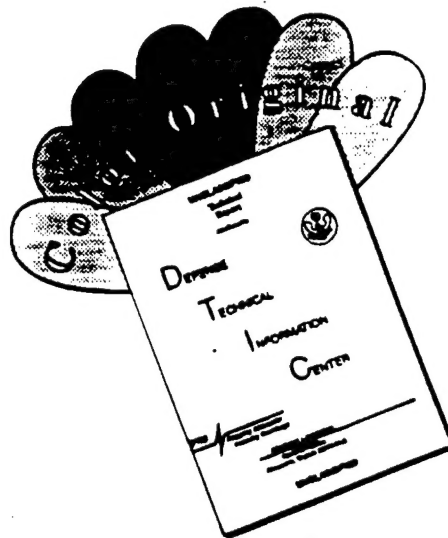


GARY D. BARMORE, Major, USAF  
Deputy Director of Surveillance & Photonics

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify Rome Laboratory/ (OCSM), Rome NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

# DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE July 1996		3. REPORT TYPE AND DATES COVERED Final Sep 95 - Mar 96	
4. TITLE AND SUBTITLE MULTISENSOR TRACK-TO-TRACK FUSION FOR AIRBORNE SURVEILLANCE SYSTEMS				5. FUNDING NUMBERS C - F30602-95-C-0288 PE - 65502F PR - STTR TA - 51 WU - 04	
6. AUTHOR(S) R.K. Saha (Nova Research Corporation), K.C. Chang (George Mason University), and M.M. Kokar (Northeastern Univ.)					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Nova Research Corporation 203 Middlesex Turnpike, Suite #6 Burlington MA 01803-3383				8. PERFORMING ORGANIZATION REPORT NUMBER  N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory/OCSM 26 Electronic Pky Rome NY 13441-4514				10. SPONSORING/MONITORING AGENCY REPORT NUMBER  RL-TR-96-123	
11. SUPPLEMENTARY NOTES  Rome Laboratory Project Engineer: Mark G. Alford/OCSM/(315) 330-3573					
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report describes a track-to-track fusion algorithm for airborne surveillance systems employing multiple dissimilar sensors (radar, IR, and laser radar). These sensors detect targets and create tracks at different data rates. The algorithm presented performs synchronization by predicting the slower tracks to the update times of the faster tracks. The synchronized tracks are then tested for association to determine whether or not the two tracks originated from the same target. It is shown that the probability distribution of correct track association can be improved if the test statistic for association incorporates cross-covariance between the two tracks. A recursive algorithm for computing the cross-covariance is obtained. In addition, a trade-off study involving the probability of correct association, number of track matching points, size of association gate, and probability of false correlation has been prepared. These algorithms are coded in MATLAB and the results of simulations confirming the proof of concept are also presented.					
14. SUBJECT TERMS C3I, Track synchronization, Track association, Track fusion, Surveillance, Target tracking				15. NUMBER OF PAGES 60	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
				20. LIMITATION OF ABSTRACT UL	

## TABLE OF CONTENTS

SECTION	PAGE
1      Introduction	1
1.1    Background	2
1.2    Approach	3
1.3    Overview	4
2      Mathematical Model	5
3      Kinematic Track Synchronization	9
4      Kinematic Track Association	10
4.1 Probability of Correct Association	10
4.2 Probability of False Correlation	14
5      Kinematic Fusion of Asynchronous Tracks	18
5.1 Description of Target and Track Models	19
5.2 Description of Weighted Covariance KTF Algorithm	20
5.3 Description of Information Fusion Approach to KTF	25
6      Simulation Results	29
6.1 Simulation of Weighted Covariance Fusion Algorithm	30
6.2 Simulation of Information Fusion Algorithm	41
6.3 Performance Comparison	42
7      Summary	47
List of References	48
Appendix A      Description of Sensor Models	50

## LIST OF FIGURES

FIGURE	PAGE
1      Airborne Surveillance Scenario	30
2      Sensor-Scenario Definition Dialog Box	32
3      Trajectory Setup Dialog Box	33
4      Laser Radar Definition Dialog Box	33
5      IR Tracks	35
6      IR Tracker Position Errors	35
7      Radar Tracks	36
8      Radar Tracker Position Errors	36
9      Laser Radar Tracks	37
10     Laser Radar Tracker Position Errors	37
11     Fused Tracks for IR and Radar	38
12     Fused Position Errors for IR and Radar	38
13     Fused Tracks for IR and Laser Radar	39
14     Fused Position Errors for IR and Laser Radar	39
15     Fused Tracks for Radar and Laser Radar	40
16     Fused Position Errors for Radar and Laser Radar	40
17     Ratios of Elements of Covariance Matrix for Fused State Estimates to Single Sensor State Estimates	45
18     Ratios of Area Ellipses for Fused State Estimates to Single Sensor State Estimates	46

## LIST OF TABLES

TABLE		PAGE
1	Probability of Correct Association Trade-Off Study	13
2	Probability of False Correlation Trade-Off Study	17

## SECTION 1

### INTRODUCTION

Design of current tactical air defense systems rely on inputs from a network of distributed sensors and/or platforms to construct an air picture and to coordinate vital mission activities such as surveillance, early warning, and weapons control. Integration of various data channels enhance effectiveness of surveillance, improve immunity to interference and deception, increase information capacity, and reduce eclipsed (where data are unavailable because of eclipsing by the transmission) and blind zones. Current system architectures, however, consist of separate track files being maintained by individual sensor platforms. This track data must be associated and fused to provide a coordinated assessment of the tactical environment. The de facto standard for the exchange of air surveillance data is through the use of track files over non-dedicated tactical data links. The contents of these messages limit the data available for track-to-track processing, and incompatible network protocols constrain the dispersion of track files among users. As a result, existing tactical surveillance systems are unable to provide an integrated air picture obtainable in a centralized fusion system architecture.

Previous work on track-to-track fusion was based on the assumption that the estimation errors of tracks for the same target obtained from different sources are uncorrelated [1]. This assumption is incorrect, however, because the process noise associated with target maneuvers is common to the filter dynamics used by the estimators [2]. Hence, the target tracks are correlated even though the sensor measurement errors are uncorrelated. The effect of the cross-covariance on the performance of track-to-track association and kinematic fusion was investigated in [3] for fusion of tracks created by two dissimilar sensors. This problem was also investigated in [4] for fusion of synchronous tracks from two identical sensors. However, results published in unclassified literature do not discuss fusion of asynchronous tracks.



The goal in this contract is to fuse asynchronous tracks which are updated at different time update rates. Specifically, a Kinematic Track Synchronization (KTS) algorithm for synchronizing fast and slow tracks was developed. Then the minimum closeness score criteria was used in the Kinematic Track Association (KTA) algorithm to test the hypothesis that two candidate tracks for fusion originated from the same target. The associated track pair is then combined using a Kinematic Track Fusion (KTF) algorithm

## **1.1 BACKGROUND**

Most currently fielded surveillance systems employ ad hoc methods for correlating tracks from multiple sources. These methods are dependent upon specific system configurations and often lack rigorous theoretical foundations. Typically, little is done to fuse tracks which originate from the same target. Rather, a preferred source of track data is usually identified, and the tracks from that source are used to represent the correlated aggregates, while the rest of the track file is ignored. A track quality (TQ) parameter provided by each track file is often the discriminant used for selecting a preferred track source. Unfortunately, there is no rigorous criteria for determining TQ, instead, all systems are free to compute TQ as desired. This processing approach is adopted because of constraints imposed by the requirement to adhere to existing tactical data link protocols

The Air Force Scientific Advisory Board Summer Panel for 1991 recommended a theater wide information exchange architecture for distributing tactical surveillance data in track form. To accomplish this task, a new breed of real time RF links is being planned to link a vast array of platforms, e.g., E-3 AWACS, E-2C Hawkeye, RC-135 Rivet Joint, Common Sensor Platforms (RC-12 Guardrail, EP-3 and ES-3A), E-8C Joint STARS, FLIR/TV-equipped UAVs, etc. This new breed of battlefield network links sensors and weapons to make it possible not only to launch missiles or vector aircraft against targets far beyond the range of each weapon's own guidance systems, but also enables sensors/ platforms to share information and automatically update the information in a multitude of networked fire control computers. Put simply, this kind of real-time C<sup>3</sup>I systems not only could coordinate sensors and shooters, but they also could provide a rich easily accessible body

of tactical information, updated in real time, to give every command center a coherent situational picture of a conflict.

## **1.2 APPROACH**

This study pursues the development of algorithms which can integrate all available track information into composite tracks of greater quality than the constituent parts. The principle thrust of this contract is the derivation of track synchronization, track-to-track association, and track fusion algorithms.

In a surveillance system, targets are detected by different sensors which exploit different characteristics of the optical, infrared, and electromagnetic spectrum. After detection, report to track association is performed and a tracker is employed to create tracks. These tracks are created at different rates which depend upon the sensor characteristics. When these asynchronous tracks arrive at the fusion center, there is a need to synchronize them prior to fusion. In this report, Kinematic Track Synchronization (KTS) is performed by predicting the slower tracks to the time of arrival of faster tracks. In addition, covariance of the predicted track is also computed which is used in the track association and fusion algorithm.

A difficult aspect of track level fusion is the decision process whereby tracks from different sources are determined to represent the same target. In this report, Kinematic Track Association (KTA) is performed by employing a track matching algorithm. This algorithm employs an association gate around faster tracks and performs a chi-square test. Slower tracks passing this test are then used to compute closeness score. The track pair which has the minimum closeness score is then selected for association.

In this report, it is assumed that all remote tracks are transmitted to a centralized fusion center. At the fusion center, Kinematic Track Fusion (KTF) is performed and the central track library is updated by the newly created fused track. In this report, track fusion is performed by employing a weighted covariance algorithm [3]. For two-sensor track fusion, this algorithm combines associated tracks by weighting each of these tracks by certain covariances which are functions of their respective filtering

variances as well as their cross-covariance. An algorithm for computing the cross-covariance between asynchronous tracks is also derived.

The KTS, KTA, and KTF algorithms developed in this contract are implemented in MATLAB. These algorithms are tested by creating an Airborne Surveillance Scenario which consists of four targets which are flying at a speed of 0.31 km/sec. maintaining a constant altitude of 9.1 km. These targets are sensed by three sensors (Radar, IR, and Laser Radar) which are colocated and stationary at 9.1 km. Sensors and the scenario are simulated using MATLAB. A Graphical User Interface (GUI) is also designed in Visual Basic.

### 1.3 OVERVIEW

This report is organized as follows. In section 2 mathematical model for track fusion is discussed. The KTS algorithm is described in section 3. In section 4 details of the KTA algorithm is given. Closed-form mathematical expressions for the probabilities of correct association and false correlation are derived. In addition, trade-off studies involving several system parameters are also presented. Description of the KTF algorithm is given in section 5. This algorithm incorporates cross-covariance between asynchronous tracks which originated from the same target. In section 6 details of numerical simulation of the fusion algorithm for an airborne surveillance scenario are presented. Finally, summary of the project and some concluding remarks are contained in section 7.

## SECTION 2

### MATHEMATICAL MODEL

For the sake of simplicity, it is assumed that two sensors of dissimilar quality (characterized by differences in their measurement variances) are tracking the same target. After acquisition, each target is tracked with a Kalman filter associated with the sensor. The mathematical model describing the target dynamics is assumed to be linear and of the form

$$\begin{aligned}
 X(t_{k+1}) &= \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} X(t_k) + \begin{bmatrix} \Delta T^2 / 2 & 0 & 0 \\ 0 & \Delta T^2 / 2 & 0 \\ 0 & 0 & \Delta T^2 / 2 \\ \Delta T & 0 & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T \end{bmatrix} W(t_k) \\
 &= \Phi X(t_k) + G W(t_k)
 \end{aligned} \tag{2-1}$$

where, the target state vector can be expressed by a six dimensional vector

$X = [u^T, v^T]^T$  with the three dimensional position vector  $u$  and three dimensional velocity vector  $v$ .

The input noise  $W(t_k)$  is assumed to be a three dimensional zero mean Gaussian noise vector whose variance is  $Q = qI$  where,  $I$  is a 3x3 identity matrix and

$$GE[W(t_k)W(t_k)^T]G^T = GQG^T = \begin{bmatrix} \frac{\Delta T^3}{3} qI & \frac{\Delta T^2}{2} qI \\ \frac{\Delta T^2}{2} qI & \Delta T qI \end{bmatrix} \tag{2-2}$$

where,  $E[.]$  denotes statistical expectation. This target is tracked by three sensors whose measurement model is described by

$$Z^i(t_k) = h^i(X(t_k)) + V^i(t_k) \tag{2-3}$$

where, the superscript  $i$  denotes the type of sensor. In this report, three different sensors (Radar, IR, and Laser Radar) are used to track the target. The observation vector  $Z^i(t_k)$  for Radar is  $[R, A, E]$ , for IR is  $[A, E]$  and for Laser Radar is  $[R, \dot{R}, A, E]$  where,  $A$  denotes Azimuth,  $E$  denotes Elevation,  $R$  denotes Range, and  $\dot{R}$  denotes Range Rate. First partial derivatives of the observation matrix  $h(X(t_k))$ , are given below:

For radar sensor,

$$H^R = \begin{bmatrix} \cos \bar{A} \cos \bar{E} & \sin \bar{A} \cos \bar{E} & \sin \bar{E} & 0 & 0 & 0 \\ -\sin \bar{A} / \bar{R} \cos \bar{E} & \cos \bar{A} / \bar{R} \cos \bar{E} & 0 & 0 & 0 & 0 \\ -\cos \bar{A} \sin \bar{E} / \bar{R} & -\sin \bar{A} \sin \bar{E} / \bar{R} & \cos \bar{E} / \bar{R} & 0 & 0 & 0 \end{bmatrix}$$

where,  $(\bar{R}, \bar{A}, \bar{E})$  is the predicted measurement vector.

For IR sensor,

$$H^I = \begin{bmatrix} -\sin \bar{A} / \bar{R} \cos \bar{E} & \cos \bar{A} / \bar{R} \cos \bar{E} & 0 & 0 & 0 & 0 \\ -\cos \bar{A} \sin \bar{E} / \bar{R} & -\sin \bar{A} \sin \bar{E} / \bar{R} & \cos \bar{E} / \bar{R} & 0 & 0 & 0 \end{bmatrix}$$

where,  $(\bar{A}, \bar{E})$  is the predicted measurement vector.

For Laser Radar,

$$H^L = \begin{bmatrix} \cos \bar{A} \cos \bar{E} & \sin \bar{A} \cos \bar{E} & \sin \bar{E} & 0 & 0 & 0 \\ -\bar{R} \cos \bar{A} \sin \bar{E} / \bar{R} & -\bar{R} \sin \bar{A} \sin \bar{E} / \bar{R} & \bar{R} \sin \bar{E} / \bar{R} & \cos \bar{A} \cos \bar{E} & \sin \bar{A} \cos \bar{E} & \sin \bar{E} \\ -\sin \bar{A} / \bar{R} \cos \bar{E} & \cos \bar{A} / \bar{R} \cos \bar{E} & 0 & 0 & 0 & 0 \\ -\cos \bar{A} \sin \bar{E} / \bar{R} & -\sin \bar{A} \sin \bar{E} / \bar{R} & \cos \bar{E} / \bar{R} & 0 & 0 & 0 \end{bmatrix}$$

where,  $(\bar{R}, \dot{\bar{R}}, \bar{A}, \bar{E})$  is the predicted measurement vector. The measurement noise vector for the three sensors are assumed to be zero mean white Gaussian noise, whose variances are described by

$$R^R = \text{diag}(\sigma_R^2, \sigma_A^2, \sigma_E^2), R^I = \text{diag}(\sigma_A^2, \sigma_E^2), R^L = \text{diag}(\sigma_R^2, \sigma_A^2, \sigma_E^2)$$

The state estimates, filter gains and covariances of the tracks are obtained from the following Extended Kalman Filter (EKF) equations:

$$\hat{X}^m(t_{k+1}/t_{k+1}) = \phi \hat{X}^m(t_k/t_k) + K^m(t_{k+1})[Z^m(t_{k+1}) - H^m(t_{k+1})\phi \hat{X}^m(t_k/t_k)] \quad (2-4)$$

$$K^m(t_{k+1}) = P^m(t_{k+1}/t_k) H^{mT}(t_{k+1}) [H^m(t_{k+1}) P^m(t_{k+1}/t_k) H^{mT}(t_{k+1}) + (\sigma_X^m)^2]^{-1} \quad (2-5)$$

$$P^m(t_{k+1}/t_k) = \phi P^m(t_k/t_k) \phi^T + GQG^T \quad (2-6)$$

$$P^m(t_{k+1}/t_{k+1}) = [I - K^m(t_{k+1}) H^m(t_{k+1})] P^m(t_{k+1}/t_k) \quad (2-7)$$

where, the superscript  $m$  denotes the type of sensor which created the track and it is assumed that the filter dynamics are the same as that of the target. It is also assumed that these tracks are created at the remote stations every  $\Delta T$  sec. and transmitted to the fusion center over a data link at the same rate. Time lags and delays are neglected. The input noise  $W(t_k)$  introduces cross-correlation between the trackers which are tracking the same target. Effect of the cross-correlation between synchronous tracks is reflected in the cross-covariance matrix, denoted by  $P^c$ , which is given [2] by

$$\begin{aligned} P^c(t_k/t_k) &= E[X(t_k) - \hat{X}^i(t_k)][X(t_k) - \hat{X}^j(t_k)]^T \\ &= [I - K^i(t_k) H^i] \phi P^c(t_{k-1}/t_{k-1}) \phi^T [I - K^j(t_k) H^j]^T \\ &\quad + [I - K^i(t_k) H^i] GQG^T [I - K^j(t_k) H^j]^T \end{aligned} \quad (2-8)$$

with  $P^c(0/0) = 0$ . An estimate<sup>1</sup> of the fused track and its covariance, respectively, is given by [3]

$$\hat{X}^F(t_k/t_k) = \hat{X}^i(t_k/t_k) + [P^i(t_k/t_k) - P^c(t_k/t_k)](P^E)^{-1}[\hat{X}^j(t_k/t_k) - \hat{X}^i(t_k/t_k)] \quad (2-9)$$

<sup>1</sup> Recently, it has been shown [6] that this estimate is not optimal.

$$P^F(t_k / t_k) = P^i(t_k / t_k) - [P^i(t_k / t_k) - P^c(t_k / t_k)](P^E)^{-1}[P^i(t_k / t_k) - P^c(t_k / t_k)]^T \quad (2-10)$$

where

$$P^E(t_k / t_k) = P^i(t_k / t_k) + P^j(t_k / t_k) - P^c(t_k / t_k) - P^c(t_k / t_k)^T \quad (2-11)$$

At steady state, equation (2-8) can be written as

$$P^c = F_1 P^c F_2^T + Q^* \quad (2-12)$$

where,

$$F_1 = [I - K^i H^i] \phi$$

$$F_2 = [I - K^j H^j] \phi \quad (2-13)$$

and the Kalman filter gains for the sensors are described by

$$K^m = \begin{bmatrix} K_1^m \\ K_2^m \end{bmatrix} \quad (2-14)$$

In addition, the input matrix  $Q^*$ , shown in (2-12) is given by

$$Q^* = [I - K^i H^i] G Q G^T [I - K^j H^j]^T \quad (2-15)$$

Equation (2-12) is an asymmetric discrete Lyapunov equation whose steady state solution was obtained by means of matrix inversion in [5].

## SECTION 3

### KINEMATIC TRACK SYNCHRONIZATION

In today's Command, Control, Communication, and Intelligence (C<sup>3</sup>I) environment, remote stations are required to transmit their local tracks instead of raw measurements, to a fusion center over a communication link. The communication links have different channel capacity, bandwidth, and data rates. Hence the remote tracks arrive at the fusion center at different rates. The track fusion algorithm implemented in this contract, takes into account the rates of transmission for each link. Specifically, slower tracks are predicted to the arrival time of the faster tracks. In addition, covariance of the slower track is updated by propagating it as follows:

$$\begin{aligned}\hat{X}^i(t_{k+m} / t_k) &= \phi^m \hat{X}^i(t_k / t_k) \\ \text{and} \\ P^i(t_{k+m} / t_k) &= \phi^m P^i(t_k / t_k) (\phi^m)^T + \sum_{j=1}^m \phi^{j-1} G Q G^T (\phi^{j-1})^T\end{aligned}\tag{3-1}$$

where, the superscript  $i$  denotes the track number,  $t_{k+m} = (k+m)\Delta T$ , and  $\phi^m = \phi^{m-1} \cdot \phi$ . It is assumed that the track  $i$  was last updated at time  $t_k$ . When this track is associated with another track  $j$  at time  $t_{k+m}$ , the association algorithm takes into account the updated covariance of the propagated tracks as shown in (3-1). Detail of the association algorithm is described in the next section. If the association algorithm accepts the hypothesis that the two tracks originated from the same target, then the track fusion algorithm combines the track  $j$  and the propagated track  $i$ . Detail of the track fusion algorithm are given in section 5.



## SECTION 4

### KINEMATIC TRACK ASSOCIATION

The Kinematic Track Association (KTA) algorithm implemented in this contract involves testing the hypothesis (at every time update  $t_k$ ) that the two tracks  $\hat{X}^i$  and  $\hat{X}^j$  originated from the same target [7]. Hence, the null hypothesis can be stated as:  $H_0: \hat{X}^i - \hat{X}^j = 0$  vs.  $H_1: \hat{X}^i - \hat{X}^j \neq 0$ . If the tracks are independent [1], then under hypothesis  $H_0$ ,  $\text{cov}(\hat{X}^i - \hat{X}^j) = P^i + P^j$ . Assuming Gaussian probability distribution of the track estimates, the test of hypothesis can be restated as:

$$H_0: (\hat{X}^i - \hat{X}^j)^T (P^i + P^j)^{-1} (\hat{X}^i - \hat{X}^j) \leq \lambda_n \quad (4-1)$$

where,  $\lambda_n$  denotes the gate width of the association region. Since the test statistic  $(\hat{X}^i - \hat{X}^j)^T (P^i + P^j)^{-1} (\hat{X}^i - \hat{X}^j)$  is chi-square with  $n$  degrees of freedom, the test threshold  $\lambda_n$  can be chosen for a level of significance  $\alpha$  such that

$$\Pr\{(\hat{X}^i - \hat{X}^j)^T (P^i + P^j)^{-1} (\hat{X}^i - \hat{X}^j) > \lambda_n / H_0\} = \alpha \quad (4-2)$$

#### 4.1 PROBABILITY OF CORRECT ASSOCIATION

For every track pairs within an association gate, let  $N$  denote the total number of common track points and let  $M$  denote the total number of times both the tracks were updated simultaneously, so that  $N-M$  equals total number of times at least one track was not updated. Then the test statistic for evaluating how close the two tracks are, over the entire track matching interval, can be obtained by constructing the closeness score,  $CS(t_M)$  defined as:

$$CS(t_M) = \sum_{k=1}^M (\hat{X}^i(t_k / t_k) - \hat{X}^j(t_k / t_k))^T [P^i(t_k / t_k) + P^j(t_k / t_k)]^{-1} (\hat{X}^i(t_k / t_k) - \hat{X}^j(t_k / t_k)) \quad (4-3)$$

which is a cumulative sum over M track match points. The probability distribution of correct association of two tracks  $\hat{X}^i$  and  $\hat{X}^j$ , denoted by  $P_{CA}$  is then obtained by assuming independence of the error between the two track estimates and by testing  $CS(t_M)$  against the threshold  $\lambda_{nM}$  corresponding to the chi-square distribution with nM degrees of freedom:

$$P_{CA} = \Pr\{CS(t_M) \leq \lambda_{nM} / H_0\} = 1 - \alpha \quad (4-4)$$

In order to incorporate the dependence between the track estimates because of the common process noise, the closeness score is modified as:

$$CS^*(t_M) = \sum_{k=1}^M \{\hat{X}^i(t_k / t_k) - \hat{X}^j(t_k / t_k)\}^T P^E(t_k / t_k)^{-1} \{\hat{X}^i(t_k / t_k) - \hat{X}^j(t_k / t_k)\} \quad (4-5)$$

Let  $CS_1(t_M) = \theta CS^*(t_M) / N$ , where,  $\theta$  is a sensor dependent constant. Then at every time update, the closeness score is converted to a Figure of Merit (FOM) by mapping it to a range [0,1] by means of the following transformation:

$$FOM(t_M) = \frac{1}{1 + CS_1(t_M)} = \frac{b}{b + CS^*(t_M)} \quad (4-6)$$

where,  $b = N / \theta$ . This FOM is also used as a goodness-of-fit statistic for track-to-track association. Incorporating the cross-covariance in the modified closeness score, the probability distribution of correct association denoted by  $P_{CA}^*$  becomes

$$P_{CA}^* = \Pr\{CS^*(t_M) \leq \lambda_{cs^*, nM}^* / H_0\} = 1 - \alpha^* \quad (4-7)$$

A measure of effectiveness which was found useful for this track-to-track association problem is the FOM, given in (4-6). Corresponding to this FOM, the probability of correct association,  $P_{CA}^*$  is the cumulative probability distribution function given by:

$$\begin{aligned}
 P_{CA}^* &= \Pr(\text{FOM} \leq \lambda_{\text{FOM}, nM}) \\
 &= \Pr\left(\frac{b}{b + CS^*(t_M)} \leq \lambda_{\text{FOM}, nM}\right) \\
 &= 1 - \int_0^{\lambda_{cs^*, nM}} f_{cs^*, nM}(z) dz
 \end{aligned} \tag{4-8}$$

where,  $f_{cs^*, nM}(z)$  is the chi-square density with  $nM$  degrees of freedom, and  $\lambda_{cs^*, nM}$  is given by

$$\lambda_{cs^*, nM} = b\left(\frac{1}{\lambda_{\text{FOM}, nM}} - 1\right) \tag{4-9}$$

It is to be noted that numerical computation for  $P_{CA}^*$  using (4-8) is complicated. Instead, an indirect approach using (4-9) is preferable because it lends itself to the use of readily available tables for chi-square densities. Using such a table, trade-off studies for a goodness-of-fit test can be readily performed, as shown in table 1.

**Table 1. Probability of Correct Association Trade-Off Study**

$P_{CA}$	M	$\lambda_{CS^*,nM}$	b	$\lambda_{FOM,nM}$
$\Pr(FOM \leq \lambda_{FOM,nM}) = 0.90$	2	6.30	119.70	0.95
			623.70	0.99
	3	10.9	207.10	0.95
			1079.10	0.99
	5	20.60	391.40	0.95
			2039.40	0.99
$\Pr(FOM \leq \lambda_{FOM,nM}) = 0.95$	2	5.23	99.37	0.95
			517.70	0.99
	3	9.40	178.60	0.95
			930.60	0.99
	5	18.50	351.50	0.95
			1831.50	0.99

This table is useful for a system designer who may choose to select the level of  $P_{CA}$  a-priori. For example, if it is desired to have  $P_{CA} = 0.95$ , then (4-8) is used to fix the threshold,  $\lambda_{cs^*,nM}$ , for each  $M$ , which corresponds to a level of significance of 0.05. In this study, it is assumed that the three sensors are equipped with six dimensional Kalman filters. Hence,  $n = 6$ . At the end of every fast track update time  $t_k$ , the slower tracks are predicted to  $t_k$ . The predicted tracks are subjected to a chi-square test for association using six degrees of freedom. Every time the predicted track passes this test, the counter for  $N$  is incremented by 1 and the number of track match points  $M$  is adjusted. At time  $t_k$  if the number of track matching points increase, then the counter for  $M$  is incremented by 1, otherwise, it is left unchanged. Using this value of  $M$ , for each fast track  $i$  and every slow track  $j$ , the closeness score,  $CS^*(t_M)$ , is computed using (4-5). For each  $t_k$ , these values of  $CS^*(t_M)$  are compared with  $\lambda_{cs^*,nM}$  which can be obtained from a look-up table. Final selection for association is made for the track pair which has the highest level of  $CS^*(t_M)$  and exceeds  $\lambda_{cs^*,nM}$ .

Table 1 also shows a trade-off between  $b$  and  $\lambda_{FOM,nM}$ . In addition, the choice of  $b$  affects the parameter  $\theta$  as shown in (4-6). But the choice of  $\theta$  is dependent upon the sensor mode and other parameters. Hence, there is a trade-off between  $\lambda_{FOM,nM}$  and  $\theta$  as shown in table 1.

## 4.2 PROBABILITY OF FALSE CORRELATION

Selection of the threshold for the test of association has direct bearing upon the probability of false correlation  $P_{FC}$ . Analytical derivation of this expression in closed-form follows closely the methodology developed in [8]. In this analysis, it is assumed that: a) false tracks are uniformly distributed over the six degree of freedom chi-square gate for association, and b) false correlations are independent for each update which, essentially, implies random clutter.

Let  $\lambda_g$  denote the size of the six degree of freedom chi-square gate for association. This gate size must be selected to let sufficient number of slow tracks through this window. Assuming that a false track is uniformly distributed within this gate, the cumulative probability distribution function of

$$y(t_k) = [\hat{X}^i(t_k / t_k) - \hat{X}^j(t_k / t_k)]^T (P^E(t_k))^{-1} [\hat{X}^i(t_k / t_k) - \hat{X}^j(t_k / t_k)] \quad (4-10)$$

is given by

$$\Pr[y(t_k) \leq \lambda] = \left(\frac{\lambda}{\lambda_g}\right)^{\frac{n}{2}} \quad \text{for } 0 \leq \lambda \leq \lambda_g \quad (4-11)$$

thus, the pdf of  $y(t_k)$  is

$$f[y(t_k)] = \frac{n}{2} \frac{(y)^{\frac{n}{2}-1}}{(\lambda_g)^{\frac{n}{2}}} \quad \text{for } 0 \leq \lambda \leq \lambda_g \quad (4-12)$$

Assuming independence of false association at each  $t_k$ , the joint pdf of  $M$  false correlations is the convolution of  $M$  pdf's given by (4-12). Hence, given a test threshold  $\lambda_{FC}$ , the false track correlation probability is

$$P_{FC} = \int_0^{\lambda_{FC}} g(x) dx \quad (4-13)$$

where,  $g(x)$  is the convolution of  $M$  pdf's given in (4-12). Evaluation of (4-13) in closed-form is analytically intractable because of the nature of  $f(y(t_k))$  given in (4-12). For this reason,  $g(x)$  is approximated by a Gaussian pdf whose mean and variance are given below

$$\mu = ME[y] = \frac{Mn\lambda_g}{n+2} \quad (4-14)$$

and

$$\sigma^2 = MVar[y] = \frac{4Mn\lambda_g^2}{(n+4)(n+2)^2} \quad (4-15)$$

Therefore, the false track correlation probability is approximated by

$$P_{FC} \approx \int_{-\infty}^{\lambda_{FC}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \quad (4-16)$$

where,

$$\lambda_{FC} = \frac{\lambda_{cs} \cdot nM - \mu}{\sigma} \quad (4-17)$$

Based on this approximation, table 2 presents results of trade-off study for  $n = 6$ , which involves  $P_{CA}$ ,  $\lambda_g$ ,  $M$ , and  $P_{FC}$ . This table shows that for a chosen  $\lambda_g$ ,  $P_{FC}$  decreases as  $M$  (the number of track matching points) increases, for a desired level of  $P_{CA}$ . On the other hand, if it is desired to hold  $P_{FC}$  to a low value, then either  $M$  must be increased, which will delay decision on track pair association and may cause track staleness, or  $P_{CA}$  must be decreased. The same results could also be achieved by making the gate window  $\lambda_g$  tighter which would, in turn, reject good candidate tracks for association, and force the system designer to reduce  $P_{CA}$ .

It is to be noted that the analysis presented in this section is an analytical tool that can be used off-line, to predict the Measure of Effectiveness (MOE) of a track fusion algorithm without resorting to elaborate Monte Carlo simulation. For this reason, the algorithms for computing probabilities of correct association and false correlation have not been implemented in the software developed in this contract.

**Table 2. Probability of False Correlation Trade-Off Study**

M	$P_{CA} = 0.90$		$P_{CA} = 0.95$	
	$\lambda_g$		$\lambda_g$	
	90%	95%	90%	95%
	$P_{FC}$	$P_{FC}$	$P_{FC}$	$P_{FC}$
1	0.90	0.68	0.99	0.90
2	0.81	0.46	0.96	0.73
3	0.72	0.29	0.92	0.54
5	0.54	0.10	0.81	0.27



## SECTION 5

### KINEMATIC FUSION OF ASYNCHRONOUS TRACKS

In this section, the algorithm for Kinematic Track Fusion (KTF) is described. The KTF algorithm is basically a weighted covariance fusion algorithm where the candidate tracks for fusion are weighted by certain covariances. The weighted tracks are then summed to obtain the fused track as shown in (2-9) - (2-10). It is assumed that targets are detected by three sensors (Radar, IR, and Laser Radar). Details of the sensor models are described in Appendix A. After detection, the sensors perform report to track association and transmit these tracks to a central station by communication links (e.g., TADIL A, B, or J). It is assumed that the Radar transmits the tracks at a rate of 4 sec., the IR sensor transmits at a rate of 2 sec. and the Laser radar transmits at a rate of 1 sec. These sensors have non-unity probability of detection ( $P_D < 1$ ). Hence, the tracks created after performing report to track association, may contain gaps due to missed detection. In addition, false tracks may be created which will result in the number of tracks being greater than the number of targets. These tracks are processed at the central station where track synchronization (described in Section 3) and association (described in Section 4) are performed. Tracks are associated by minimizing closeness score, and the track pair that gives rise to minimum closeness score are then fused using the Kinematic Track Fusion (KTF) algorithm described in this section. Track fusion is useful for airborne surveillance and  $C^3I$ , because two sensors tracking the same target may contain different kinds of information about the target.

It is well known [2] that when two sensors track the same target, the tracks become correlated due to target maneuver noise which is the same for the two trackers tracking the target. It has also been shown that, incorporation of cross-correlation between tracks in the track fusion algorithm [3] results in reduction of variance of position and speed which are assumed to be the states of the two track filters. However, published results in open literature consider fusion of synchronous tracks that are created by perfect sensors of  $P_D = 1$ . The KTF algorithm described in this section, consider fusion of asynchronous tracks that contain missing data (introduced by

sensors that have non-unity probability of detection). An efficient algorithm for computing cross-correlation between these tracks is derived. At any time  $t_k$ , if a track from one sensor is not updated, then this track is propagated to  $t_k$  from its last update at  $t_{k-1}$  and cross-correlation between this propagated track and tracks from other sensors are computed. For asynchronous tracks, this cross-correlation is computed by modifying the equation for cross-covariance given in (2-8). Then the fused track and its covariance is computed by incorporating the cross-covariance in the track fusion algorithm as shown in (2-9) and (2-10).

## 5.1 DESCRIPTION OF TARGET AND TRACK MODELS

In this section, it is assumed that four targets are moving at the same speed of 0.31 km/sec at a constant altitude of 9.10 km. The target dynamic is modeled by means of a state variable model which has six states (three dimensional positions and speed). Target maneuver noise is modeled by a three dimensional white Gaussian noise vector. This target model is updated every one second or,  $\Delta T = 1$ . Sensor accuracy is modeled by white Gaussian noise vectors. All the sensors are assumed to be colocated at the same altitude as the targets.

At the fusion center, tracks from one sensor are associated with tracks from the other sensor, one at a time. Let the tracks from one sensor be denoted by  $i$  and tracks from the other sensor be denoted by  $j$ . Tracks from sensor  $i$  (or  $j$ ) are classified as follows:

Updated tracks from sensor  $i$  at time  $t_{k+1}$  are denoted by  $i_1$  and are counted as  $NIU(t_{k+1})$ . These updated tracks are further subdivided as follows:

- $i_1$  tracks updated at  $t_k$  and at  $t_{k+1}$  are counted as  $NIU1(t_{k+1})$  and labeled as 1
- $i_1$  tracks not updated at  $t_k$  but updated at  $t_{k+1}$  are counted as  $NIU2(t_{k+1})$  and labeled as 2
- newly acquired updated tracks at  $t_{k+1}$  are counted as  $NIU3(t_{k+1})$  and labeled as 3

After a new track is received at the fusion center, it is correlated with tracks from the other sensor and then added to the existing  $NIU1(t_{k+1})$  track file. Separate processing of newly

acquired track is necessary because the initial value for the recursive cross-covariance matrix for this track is zero. Once the cross-covariance matrix is initiated, the newly acquired track can be classified as an  $i_1$  track. Hence,

$$NIU(t_k) = NIU1(t_k) + NIU2(t_k) + NIU3(t_k)$$

Non-updated tracks at  $t_{k+1}$  are denoted by  $i_2$  and are counted as  $NIXU(t_{k+1})$ . These non-updated tracks are further subdivided as follows:

- $i_2$  tracks updated at  $t_k$  but not updated at  $t_{k+1}$  are counted as  $NIXU1(t_{k+1})$
- $i_2$  tracks not updated at  $t_k$  and not updated at  $t_{k+1}$  are counted as  $NIXU2(t_{k+1})$

Non-updated  $NIXU2$  tracks are deleted if they are not updated for ten successive update times. Otherwise, computer resources necessary to implement this algorithm will be excessive.

Furthermore, because of the recursive nature of the fusion algorithm, it is necessary to store track statistics only for two successive time updates which reduces the burden of computation.

Hence, total numbers of non-updated tracks at any time  $t_{k+1}$  are:

$$NIXU(t_{k+1}) = NIXU1(t_{k+1}) + NIXU2(t_{k+1})$$

Format of a track file from sensor  $i$  is given below:

$i(\text{track \#}, \text{track id(1or2or 3) at } t_{k+1}, \text{current time}(t_{k+1}), \text{track id(1or2) at } t_k, t_k)$

Since the tracks are asynchronous, let the update rate for tracks from sensor  $i$  be denoted by  $NI$  and the update rate for tracks from sensor  $j$  be denoted by  $NJ$ . Let the minimum update rate be denoted by  $NMIN = \text{Min}(NI, NJ)$ . Tracks from these sensors are used to perform Kinematic Track Association (KTA) by minimizing closeness score, and the tracks which give rise to minimum closeness score are fused by means of a Kinematic Track Fusion (KTF) algorithm.

## 5.2 DESCRIPTION OF WEIGHTED COVARIANCE KTF ALGORITHM

At time  $t_k$ , tracks from the  $i$  and  $j$  sensors may have different classification according to the state of update (Track ID = 1 if it is updated at  $t_k$  and at  $t_{k-1}$ , Track ID = 2 if it was updated at  $t_{k-1}$  but

is not updated at  $t_k$  and Track ID = 3 if it is a newly acquired track). Hence, the algorithm for KTA must incorporate this information at every update time. Since the KTA algorithm described in Section 4, uses cross-covariance information for track matching, cross-correlation between tracks of different classification are computed separately. Derivations of these cross-correlation matrices are described below. Let

$$QI = \sum_{p=1}^{NI} \phi^{p-1}(1) G(1) Q G^T(1) (\phi^{p-1}(1))^T \quad (5-1)$$

$$QJ = \sum_{p=1}^{NJ} \phi^{p-1}(1) G(1) Q G^T(1) (\phi^{p-1}(1))^T \quad (5-2)$$

and,

$$Q2 = \sum_{p=1}^{NMIN} \phi^{p-1}(1) G(1) Q G^T(1) (\phi^{p-1}(1))^T \quad (5-3)$$

and let the recursive equation for cross-covariance be initialized at time  $t_0$  as

$P^{i,jl}(t_0, t_0; t_0, t_0) = 0$ . For a newly acquired track at time  $t_N$ , the initial cross-covariance at  $t_{N-1}$  will also be zero.

Let the tracks from sensor i be the faster tracks which are indexed by  $N_3$  and let the slower tracks from sensor j be indexed by  $N_1$ . Then  $t_{k+N_1} - t_k = \Delta T (N_1) NI = \Delta T (NJI) NJ$  where,  $N_3 = 1, 2, \dots, NJI$ . To illustrate, the tracks from Laser Radar are updated every 1 sec., or  $NJ = 1$ , and the Radar tracks are updated every 4 sec., or  $NI = 4$ . Hence,  $NJI = 4$  and tracks from both the sensors are updated simultaneously every 4 sec. For  $N_3 = 1, 2, \dots, NJI - 1$ , the Radar tracks are predicted every  $NJ = NMIN = 1$  sec. as follows:

$$\hat{X}^{jl}(t_{(N1-1)+N3} / t_{N1-1}) = \phi^{NMIN} \hat{X}^{jl}(t_{(N1-1)+(N3-1)} / t_{N1-1}) \quad (5-4)$$

and,

$$P^{jl}(t_{(N1-1)+N3} / t_{N1-1}) = \phi^{NMIN} P^{jl}(t_{(N1-1)+(N3-1)} / t_{N1-1}) (\phi^{NMIN})^T + Q2 \quad (5-5)$$

It is to be noted that for  $N_3 = NJI$ ,

$$\begin{aligned}
\hat{X}^{il}(t_{(N1-1)+N3} / t_{(N1-1)+N3}) - X(t_{(N1-1)+N3}) &= [I - K^{il}(t_{(N1-1)+N3})H^{il}(t_{(N1-1)+N3})]\phi^{NI} \\
&\quad [\hat{X}^{il}(t_{(N1-1)+N3-1} / t_{(N1-1)+N3-1}) - X(t_{(N1-1)+N3-1})] + K^{il}(t_{(N1-1)+N3})V^{il}(t_{(N1-1)+N3}) \\
&\quad - [I - K^{il}(t_{(N1-1)+N3})H^{il}(t_{(N1-1)+N3})]\left\{\sum_{p=1}^{NI}\phi^{p-1}GW(t_{(N1-1)+N3-p})\right\}
\end{aligned} \tag{5-6}$$

and,

$$\begin{aligned}
\hat{X}^{jl}(t_{(N1-1)+N3} / t_{(N1-1)+N3}) - X(t_{(N1-1)+N3}) &= [I - K^{jl}(t_{(N1-1)+N3})H^{jl}(t_{(N1-1)+N3})]\phi^{NMN} \\
&\quad [\hat{X}^{jl}(t_{(N1-1)+N3-1} / t_{(N1-1)+N3-1}) - X(t_{(N1-1)+N3-1})] + K^{jl}(t_{(N1-1)+N3})V^{jl}(t_{(N1-1)+N3}) \\
&\quad - [I - K^{jl}(t_{(N1-1)+N3})H^{jl}(t_{(N1-1)+N3})]\left\{\sum_{p=1}^{NMN}\phi^{p-1}GW(t_{(N1-1)+N3-p})\right\}
\end{aligned} \tag{5-7}$$

Since,  $t_{(N1-1)+N3} = t_{N1}$ , the cross-covariance matrix between  $\hat{X}^{il}(t_{N1} / t_{N1})$  and  $\hat{X}^{jl}(t_{N1} / t_{N1})$  is computed as,

$$\begin{aligned}
P^{il,jl}(t_{N1}, t_{N1}; t_{N1}, t_{N1}) &= [I - K^{il}(t_{N1})H^{il}(t_{N1})]\phi^{NI}P^{il,jl}(t_{N1-NJ}, t_{N1-NJ}; t_{N1-NJ}, t_{(N1-1)}) \\
&\quad (\phi^{NMN})^T [I - K^{jl}(t_{N1})H^{jl}(t_{N1})]^T + [I - K^{il}(t_{N1})H^{il}(t_{N1})]Q2[I - K^{jl}(t_{N1})H^{jl}(t_{N1})]^T
\end{aligned} \tag{5-8}$$

This recursive equation is initialized by  $P^{il,jl}(t_{N1-NJ}, t_{N1-NJ}; t_{N1-NJ}, t_{N1-1})$  which was computed at  $t_{N1-NJ}$ . Now, using (2-9) and (2-10) the fused track is computed as,

$$\begin{aligned}
P^E(t_{N1} / t_{N1}) &= P^{il}(t_{N1} / t_{N1}) + P^{jl}(t_{N1} / t_{N1}) - P^{il,jl}(t_{N1}, t_{N1}; t_{N1}, t_{N1}) - P^{il,jl}(t_{N1}, t_{N1}; t_{N1}, t_{N1})^T \\
\hat{X}^F(t_{N1} / t_{N1}) &= \hat{X}^{il}(t_{N1} / t_{N1}) + [P^{il}(t_{N1} / t_{N1}) - P^{il,jl}(t_{N1}, t_{N1}; t_{N1}, t_{N1})](P^E)^{-1} \\
&\quad [\hat{X}^{jl}(t_{N1} / t_{N1}) - \hat{X}^{il}(t_{N1} / t_{N1})] \\
P^F(t_{N1} / t_{N1}) &= P^{il}(t_{N1} / t_{N1}) - [P^{il}(t_{N1} / t_{N1}) - P^{il,jl}(t_{N1}, t_{N1}; t_{N1}, t_{N1})](P^E)^{-1} \\
&\quad [P^{il}(t_{N1} / t_{N1}) - P^{il,jl}(t_{N1}, t_{N1}; t_{N1}, t_{N1})]^T
\end{aligned} \tag{5-9}$$

Similarly, when the fast track  $i_l$  has been updated and the slow track  $j_l$  has been predicted to the update time of the fast track, cross-covariance between  $\hat{X}^{il}(t_{(N1-1)+N3} / t_{(N1-1)+N3})$  and

$\hat{X}^{jl}(t_{(N1-1)+N3} / t_{N1-1})$  for  $N_3 = 1, 2, \dots, N_{J1}-1$  can be derived as,

$$\begin{aligned}
P^{i1,j1}(t_{N1-1+N3}, t_{N1-1+N3}; t_{N1-1+N3}, t_{N1-1}) &= [I - K^{i1}(t_{N1-1+N3})H^{i1}(t_{N1-1+N3})]\phi^{N1} \\
P^{i1,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, t_{(N1-1)}) &(\phi^{NM1N})^T + \quad (5-10) \\
[I - K^{i1}(t_{N1-1+N3})H^{i1}(t_{N1-1+N3})]Q2
\end{aligned}$$

Now at every  $N_3$ , the fast track update index, the fused track and its covariance can be obtained by using (2-9) and (2-10) similar to (5-9).

Cross-correlation between the  $i_1$  tracks  $\hat{X}^{i1}(t_{(N1-1)+N3} / t_{(N1-1)+N3})$  which are updated at  $t_{(N1-1)+N3}$  and were last updated at  $t_{(N1-1)+N3-M1}$  and the  $j_1$  tracks  $\hat{X}^{j1}(t_{(N1-1)+N3} / t_{(N1-1)})$  can be derived as,

$$\begin{aligned}
P^{i1,j1}(t_{N1-1+N3}, t_{N1-1+N3}; t_{N1-1+N3}, t_{N1-1}) &= [I - K^{i1}(t_{N1-1+N3})H^{i1}(t_{N1-1+N3})]\phi^{N1} \\
P^{i2,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M1}; t_{(N1-1)+N3-1}, t_{(N1-1)}) &(\phi^{NM1N})^T + \quad (5-11) \\
[I - K^{i1}(t_{N1-1+N3})H^{i1}(t_{N1-1+N3})]Q2
\end{aligned}$$

This equation is initialized by  $P^{i2,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M1}; t_{(N1-1)+N3-1}, t_{N1-1})$  which was computed at  $t_{(N1-1)+N3-1}$ . Computation of fused tracks is now performed using (2-9) and (2-10).

Cross-correlation between  $i_2$  tracks  $\hat{X}^{i2}(t_{(N1-1)+N3} / t_{(N1-1)+N3-1})$  counted as NIXU1 which were last updated at  $t_{(N1-1)+N3-1}$  and  $j_1$  tracks  $\hat{X}^{j1}(t_{(N1-1)+N3} / t_{(N1-1)})$  is described below:

$$\begin{aligned}
P^{i2,j1}(t_{N1-1+N3}, t_{N1-1+N3-1}; t_{N1-1+N3}, t_{N1-1}) &= \phi^{N1} P^{i1,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, \\
&t_{(N1-1)}) (\phi^{NM1N})^T + Q2 \quad (5-12)
\end{aligned}$$

This equation is initialized by  $P^{i1,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, t_{N1-1})$  which was computed at  $t_{(N1-1)+N3-1}$ . Computation of fused tracks is now performed using (2-9) and (2-10).

Cross-correlation between  $i_2$  tracks  $\hat{X}^{i2}(t_{(N1-1)+N3} / t_{(N1-1)+N3-M2})$  counted as NIXU2 which were last updated at  $t_{(N1-1)+N3-M2}$  and  $j_1$  tracks  $\hat{X}^{j1}(t_{(N1-1)+N3} / t_{(N1-1)})$  is described below:

$$P^{i2,j1}(t_{N1-1+N3}, t_{N1-1+N3-M2}; t_{N1-1+N3}, t_{N1-1}) = \phi^{N1} P^{i2,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3-1}, t_{(N1-1)}) (\phi^{NMIN})^T + Q2 \quad (5-13)$$

This equation is initialized by  $P^{i2,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3-1}, t_{N1-1})$  which was computed at  $t_{(N1-1)+N3-1}$ . Computation of the cross-correlation between  $j_2$  tracks and  $i_1$  as well as  $i_2$  tracks is performed using identical methodology and further details are omitted. It is to be noted that the  $j_2$  tracks  $\hat{X}^{j2}(t_{(N1-1)+N3} / t_{(N1-2)})$  counted as NJXU1 were  $j_1$  tracks at  $t_{(N1-2)}$ . These tracks are propagated to  $t_{(N1-1)+N3}$  and then correlated with  $i_2$  tracks. The  $i_2$  tracks are processed similarly. To illustrate, cross-correlation between  $i_2$  tracks counted as NIXU1 and  $j_2$  tracks counted as NJXU1 is given by,

$$P^{i2,j1}(t_{N1-1+N3}, t_{N1-1+N3-1}; t_{N1-1+N3}, t_{N1-2}) = \phi^{N1} P^{i1,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, t_{(N1-2)}) (\phi^{NMIN})^T + Q2 \quad (5-14)$$

This equation is initialized by  $P^{i1,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, t_{N1-2})$  which was computed at  $t_{(N1-1)+N3-1}$ . Similarly, cross-correlation between  $i_2$  tracks counted as NIXU2 which were last updated at  $t_{(N1-1)+N3-M2}$  and  $j_2$  tracks counted as NJXU1 is given by,

$$P^{i2,j2}(t_{(N1-1)+N3}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3}, t_{N1-2}) = \phi^{N1} P^{i2,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3-1}, t_{(N1-2)}) (\phi^{NMIN})^T + Q2 \quad (5-15)$$

This equation is initialized by  $P^{i2,j1}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3-1}, t_{N1-2})$  which was computed at  $t_{(N1-1)+N3-1}$ . Cross-correlation between  $i_2$  tracks counted as NIXU1 which were last updated at  $t_{(N1-1)+N3-1}$  and  $j_2$  tracks counted as NJXU2 is given by,

$$P^{i2,j2}(t_{N1-1+N3}, t_{(N1-1)+N3-1}; t_{N1-1+N3}, t_{N1-M3}) = \phi^{N1} P^{i1,j2}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, t_{(N1-M3)}) (\phi^{NMIN})^T + Q2 \quad (5-16)$$

This equation is initialized by  $P^{i1,j2}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-1}; t_{(N1-1)+N3-1}, t_{N1-M3})$  which was computed at  $t_{(N1-1)+N3-1}$ . Similarly, cross-correlation between  $i_2$  tracks counted as NIXU2 which were last updated at  $t_{(N1-1)+N3-M2}$  and  $j_2$  tracks counted as NJXU2 is given by,

$$P^{i2,j2}(t_{N1-1+N3}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3}, t_{N1-M3}) = \phi^{NI} P^{i2,j2}(t_{(N1-1)+N3-1}, t_{(N1-1)+N3-M2}; t_{(N1-1)+N3-1}, t_{(N1-M3)}) (\phi^{NMIN})^T + Q2 \quad (5-17)$$

For  $N3 = NJI$ , slower tracks from sensor  $j$  are updated and cross-correlation between these updated tracks and the updated tracks from sensor  $i$  is computed. For  $i_1$  tracks counted as NIU1 and  $j_1$  tracks counted as NJU1, the cross-correlation matrix is shown in (5-8). Cross-correlation between tracks of other classifications (i.e.,  $i_2$  and  $j_1$ ,  $i_2$  and  $j_2$ , and  $i_2$  and  $j_2$ ) can be derived using the methodology described earlier.

Initialization: Three track updates from each sensor are required to initialize the KTF algorithm. Until the third track update from the slower sensor is received at the fusion center, tracks from the faster sensor are monitored and the track labels are stored. Labels of  $i_1$  tracks that have been updated at the current time update  $t_{k+1}$  and the previous time update  $t_k$  are stored as 1. The  $i_1$  tracks that have been updated at the current time update  $t_{k+1}$  but were not updated at  $t_k$  are labeled as 2. These  $i_1$  tracks were created earlier but contain missing data. The  $i_2$  tracks that were created earlier but have not been updated recently are predicted one step ahead.

This algorithm was coded in MATLAB and the simulation results are shown in Section 6. An alternative algorithm based on information fusion was also implemented. Details of this software are described in the next section.

### 5.3 DESCRIPTION OF INFORMATION FUSION APPROACH TO KTF

Recently, there have been considerable interest in the information fusion approach [9-11] to centralized KTF. This approach is different from the weighted covariance fusion described in Section 2. The difference is in the methodology employed to incorporate the track correlation



information contained in the remote tracks that are transmitted to the fusion center. In the weighted covariance track fusion methodology, the algorithm assumes that the remote tracks are correlated over time and are also correlated because of the input maneuver noise which is common to both the tracks. The KTF algorithm described in Section 2, computes cross-covariance between the tracks recursively and updates it as soon as new information is received at the fusion center. Then this algorithm weights the tracks by incorporating the cross-covariance matrix and the weighted tracks are fused using (2-9). On the other hand, the information fusion approach recognizes the fact that remote tracks are correlated in time and the measurement noise could be correlated. But this algorithm does not take into account cross-correlation between tracks.

In this section, it is shown that the KTF algorithm (2-9)-(2-10) which was derived in [3] is not exact but an approximation of the optimal solution. In fact, a published study [15] showed that the performance of the fusion algorithm derived in [3] (referred to as the "state vector fusion method") is consistently worse (biased) than the optimal method (referred to as the "measurement fusion method"). It was pointed out that the method of combining tracks using state vector fusion is, in general, suboptimal. However, this study failed to recognize that the real reason behind the performance degradation of this algorithm is an error in its derivation. In this section, the nature of the approximation made in [3] is examined. Performance of this approximate algorithm is compared with that of a track fusion algorithm based on information fusion in the next section.

The KTF algorithm (2-9)-(2-10) derived in [3] is based on the following equality

$$E[\hat{x}^j | D^i] = \hat{x}^i \quad (5-18)$$

where,  $D^i$  denotes the information (accumulated measurements) from sensor  $i$ . This equality is incorrect because

$$\begin{aligned}
E[\hat{x}^j | D^i] &= \int \hat{x}^j p(D^j | D^i) dD^j = \int E[x | D^j] p(D^j | D^i) dD^j \\
&= \int \left[ \int x p(x | D^j) dx \right] p(D^j | D^i) dD^j \\
&= \int x \int p(x | D^j) p(D^j | D^i) dD^j dx \\
&\neq \int x \int p(x | D^i, D^j) p(D^j | D^i) dD^j dx \\
&= \int x p(x | D^i) dx \\
&= E[x | D^i] = \hat{x}^i
\end{aligned} \tag{5-19}$$

Hence, it is clear that (5-18) will hold if and only if  $p(x | D^j) = p(x | D^i, D^j)$  which is incorrect. Therefore, (5-18) can only be considered an approximation and so is the KTF algorithm (2-9)-(2-10).

An alternative KTF algorithm based on the concept of information fusion [9] is described below:

$$\begin{aligned}
\hat{x}^F(t_{k+1} / t_{k+1}) &= P^F(t_{k+1} / t_{k+1}) (P^i(t_{k+1} / t_{k+1})^{-1} \hat{x}^i(t_{k+1} / t_{k+1}) + P^j(t_{k+1} / t_{k+1})^{-1} \hat{x}^j(t_{k+1} / t_{k+1}) \\
&\quad - P^i(t_{k+1} / t_k)^{-1} \bar{x}^i(t_{k+1} / t_k) - P^j(t_{k+1} / t_k)^{-1} \bar{x}^j(t_{k+1} / t_k) + P^F(t_{k+1} / t_k)^{-1} \bar{x}^F(t_{k+1} / t_k) \\
P^F(t_{k+1} / t_{k+1})^{-1} &= P^i(t_{k+1} / t_{k+1})^{-1} + P^j(t_{k+1} / t_{k+1})^{-1} - P^i(t_{k+1} / t_k)^{-1} - P^j(t_{k+1} / t_k)^{-1} + \\
&\quad P^F(t_{k+1} / t_k)^{-1}
\end{aligned} \tag{5-20}$$

This algorithm is exact if the two sensors communicate each time (full-rate communication) they receive measurements and update tracks or when the process noise is zero (deterministic system). Hence, in general, this algorithm is also approximate. It is to be noted that the time correlation information contained in the fusion algorithm (5-20) is removed by subtracting out the prior information contained in the last terms of this equation. In addition, this algorithm does not take into account the cross-correlation due to input maneuver noise. In order to compare the performance of the two algorithms, the fused covariance equation in (5-20) is rewritten as,

$$\begin{aligned}
P^F &= (P^i{}^{-1} + P^j{}^{-1} - \bar{P}^i{}^{-1} - \bar{P}^j{}^{-1} + \bar{P}^F{}^{-1})^{-1} \\
&= P^i \left[ P^i + P^j - P^j (\bar{P}^i{}^{-1} + \bar{P}^j{}^{-1} - \bar{P}^F{}^{-1}) P^i \right]^{-1} P^j \\
&= P^i - P^i (P^i + P^j - P^\#)^{-1} (P^i - P^\#)
\end{aligned} \tag{5-21}$$

and,

$$P^\# \equiv P^j (\bar{P}^i{}^{-1} + \bar{P}^j{}^{-1} - \bar{P}^F{}^{-1}) P^i$$

where,  $\bar{P}^i$ ,  $\bar{P}^j$ , and  $\bar{P}^F$  denote the predicted covariances, and for the sake of simplicity, time dependence has been omitted. It can be seen from (5-21), that one reason why the weighted covariance fusion algorithm (2-9)-(2-10) is not exact is due to the fact that no prior information is used in this formulation. It should also be noted that when the tracks from the two sensors are assumed to be independent, then the cross-correlation is zero, or,  $P^c = P^\# = 0$ . and the two fusion algorithms become identical. Furthermore, when the two sensors transmit tracks asynchronously, (5-20) also becomes approximate. This is so, because the assumption that the two sensor measurement errors are independent, becomes invalid due to the propagation of the process noise through predicted covariances[9].

This algorithm was coded in MATLAB and details of this software are described in the next section.

## SECTION 6

### SIMULATION RESULTS

This section describes the results of simulation of the two algorithms described in Section 5. A three sensor, four target scenario shown in Figure 1. was simulated using MATLAB 4.0 software package and a supporting platform. In this scenario, three sensors (Radar, IR, and Laser Radar) are colocated and mounted on a surveillance platform which is stationary at an altitude of 9.10 km. Threat aircraft altitude is also at 9.10 km. Speed of the target aircrafts are all the same at 0.31 km/sec. Threats are separated in the y-axis by 37.10 km. Separation between sensor platform and threats is 371 km. Details of the sensor models are given in Appendix A. For this simulation, a sensor's pointing is with respect to North and azimuth is measured clockwise. Monte Carlo simulation is performed to generate tracks from each sensor. These tracks are used at the fusion center by the two algorithms (covariance weighted fusion and information fusion) to perform track to track fusion.

The weighted covariance fusion algorithm can be executed through a Graphical User Interface (GUI) which is implemented in Visual Basic. Parameters of each sensor can be chosen by a user by entering the values in the screen for sensors, or by accepting the default parameters. New scenarios can be created by entering user chosen target parameters in the screen for targets or by accepting preloaded scenarios. The information fusion algorithm can be executed by means of a user-interactive set of questions and responses for selecting sensor and target parameters. The codes consists of a set of executable routines for generating tracks, performing track fusion, and for graphical display of the results. Software prototype developed in this contract, is portable to any platform supported by MATLAB, including PC and UNIX workstations.

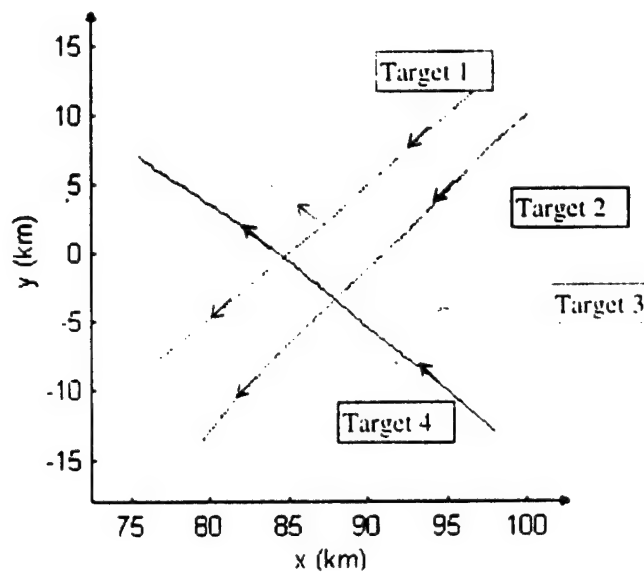


Figure 1. Airborne Surveillance Scenario

## 6.1 SIMULATION OF WEIGHTED COVARIANCE FUSION ALGORITHM

The weighted covariance KTF algorithm described in Section 5.2 is implemented in MATLAB 4.0 under Windows 3.1 environment. This software has a user-friendly interface (GUI) for creating scenarios and selecting parameters for simulating sensor models. The GUI is written in Visual Basic. The executable is MTTF.EXE. This software expects the MATLAB system be installed in the C:\MATLAB directory. It also assumes that the file MATLABRC.M is stored in that directory. Type:

```
cd c:\matlab\fusedat\demo
fusion
```

**Note:** The directory `c:\matlab\fusedat\demo` is assumed to contain the data files (input) defined by the MTTF program as well as the tracking and fusion `m`-files. This is also the recommended location for the MTTF.EXE program.

In order to run the MTTF program, follow the steps given below:

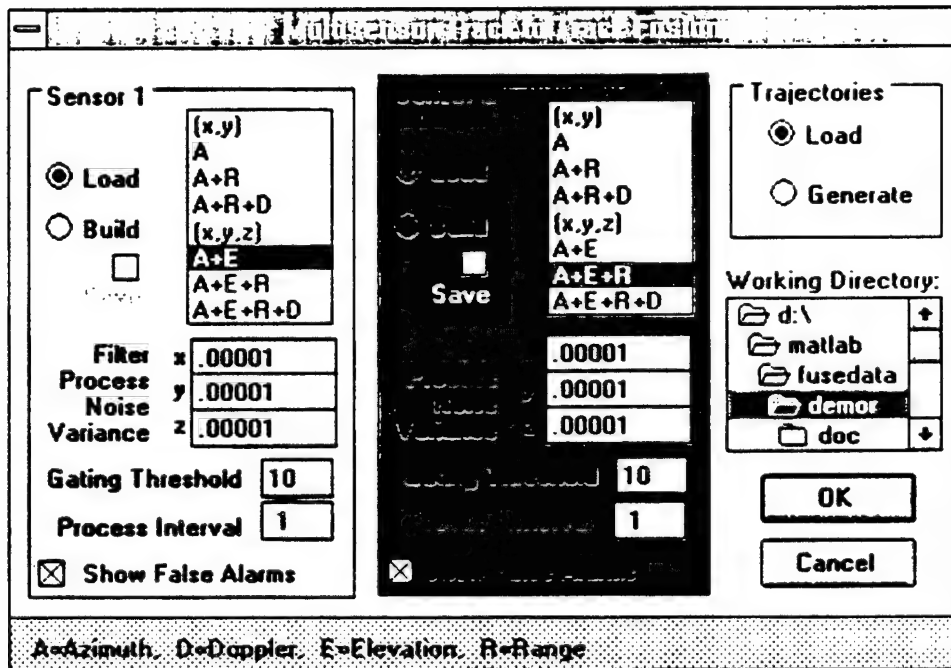
- 1 In the File Manager, open the directory **c:\matlab\fusedat\demo** and run the **MTTF.EXE** program.  
Windows opens the Multisensor Track to Track Fusion dialog box as shown in Figure 2.
- 2 To accept all default settings, click the **OK** button. All input data will be read from pre-defined files stored in the working directory (e.g. in **c:\matlab\fusedat\demo**).  
Notice that MTTF picks up initially the directory in which it is installed. If the MATLAB files are located in a different directory, select the directory using the **Working Directory** tree box.
- 3 To create custom scenarios change sensors settings in the **Sensor 1** and **Sensor 2** boxes. In order to redefine other sensor parameters the option **Build** has to be selected. To generate randomly target trajectories, in the **Trajectories** box, select the **Generate** option.

Upon pressing (clicking) the **OK** button, the program opens the **Setup for Trajectories** dialog box (if the **Generate** option is selected) as shown in Figure 3. or it goes directly to the **Sensor Type n** parameter definition dialog box (if at least one **Build** option is chosen) as shown in Figure 4.

The **Setup for Trajectories** dialog box shown in Figure 3. lets an user to determine the number of targets (default = 4), the sampling intervals (default = 10), dimension (default = 3-D), and initial position, initial velocity, and noise variance.

The **Sensor Type n** dialog box shown in Figure 4. (for Laser Radar) permits an user to define such sensor parameters as **Location**, **Accuracy**, **Minimum** and **Maximum** **Field of View**, **SNR**, **Threshold**, etc.

Upon completion of all setup tasks, MTTF invokes the MATLAB program. Through the MATLABRC.M file, the working directory is selected and the FUSION.M program started.



The dialog box is titled "Sensor-Scenario Definition Dialog Box". It is divided into several sections:

- Sensor 1:**
  - Load/Build:** Radio buttons for "Load" (selected) and "Build".
  - Filter:** A list box containing "(x,y)", "A", "A+R", "A+R+D", "(x,y,z)", "A+E", "A+E+R", and "A+E+R+D". "A+E" is currently selected.
  - Process Noise Variance:** Three input fields for x, y, and z, all containing ".00001".
  - Gating Threshold:** An input field containing "10".
  - Process Interval:** An input field containing "1".
  - Show False Alarms:** A checked checkbox.
- Scenario:**
  - Load/Build:** Radio buttons for "Load" (selected) and "Build".
  - Filter:** A list box containing "(x,y)", "A", "A+R", "A+R+D", "(x,y,z)", "A+E", "A+E+R", and "A+E+R+D". "A+E" is currently selected.
  - Process Noise Variance:** Three input fields for x, y, and z, all containing ".00001".
  - Gating Threshold:** An input field containing "10".
  - Process Interval:** An input field containing "1".
  - Show False Alarms:** A checked checkbox.
- Trajectories:**
  - Load/Generate:** Radio buttons for "Load" (selected) and "Generate".
- Working Directory:**
  - A list box showing the directory structure: "d:\", "matlab", "fusedata", "demor" (selected), and "doc".
- Buttons:** "OK" and "Cancel" buttons.

At the bottom of the dialog box, a legend states: "A=Azimuth, D=Doppler, E=Elevation, R=Range".

Figure 2. Sensor-Scenario Definition Dialog Box

Setup for Trajectories									
Initial Position									
	x	y	z						
Target 1	1	1	1						
Target 2	5	1	1						
Target 3	30	2	1						
Target 4	20	2	1						
Initial Velocity									
	vx	vy	vz						
	0.20	0.20	0.02						
	0.15	0.10	0.01						
	-0.20	0.15	0.01						
	-0.15	0.10	0.02						
Noise Variance									
	QQ(1)	QQ(2)	QQ(3)						
Target 1	0.00001	0.00001	0.00001						
Target 2	0.00001	0.00001	0.00001						
Target 3	0.00001	0.00001	0.00001						
Target 4	0.00001	0.00001	0.00001						
				Number of Targets		4	Save		<input type="checkbox"/>
				Sampling Intervals		10			
				Dimension		<input type="radio"/> 2-D <input checked="" type="radio"/> 3-D			
						OK			
						Cancel			

Figure 3. Trajectory Setup Dialog Box

Sensor Type 8 - Azimuth + Elevation + Range + Doppler				
Sensor Location		x	y	z
		0	12	0
Sensor Accuracy		Azimuth	Elevation	Range Doppler
		0	0	.02 .001
Sensor Min		-3.142	-3.142	0
Sensor Max		3.142	3.142	200
SNR [db]		100		
SNR Threshold		10		
Doppler Threshold		.001		
		OK		
		Cancel		

Figure 4. Laser-Radar Definition Dialog Box



Three routines were developed to perform Monte Carlo simulation of two-sensor track fusion. Names of these routines are: demo67, demo68, and demo78. These routines can be executed by typing, say "demo67<CR>". Details of these routines are given below. Monte Carlo simulation was performed over a time period of 100 sec. Figure 5. shows the four targets and the IR tracks of those targets. Figure 6. shows the position errors with this tracker from  $t = 0.0$  to  $t = 100$  sec. Since the 2-D IR sensor is tracking a 3-D target, position errors increase with time. Figure 7. shows the Radar tracks and the position errors with this 3-D sensor are shown in Figure 8. In this case, the position errors are less than the IR tracker errors. Figure 9. shows the Laser Radar tracks and Figure 10. shows the position errors for this tracker. As expected, position errors near target cross-over are larger than in rest of the scenario. Results of fusion of these tracks using the weighted covariance KTF algorithm are given below:

1) demo67 implements fusion of tracks from IR and Radar sensors. Figure 11. shows the fused tracks and Figure 12. shows the fusion errors. Figure 12. shows that considerable reduction in position errors can be achieved by performing track to track fusion.

2) demo68 implements fusion of tracks from IR and Laser Radar sensors. Figure 13. shows the fused tracks and Figure 14. shows the fusion errors. Figure 14. shows that considerable reduction in position errors can be achieved by performing track to track fusion.

3) demo78 implements fusion of tracks from Radar and Laser Radar sensors. Figure 15. shows the fused tracks and Figure 16. shows the fusion errors. Figure 16. shows that considerable reduction in position errors can be achieved by performing track to track fusion.

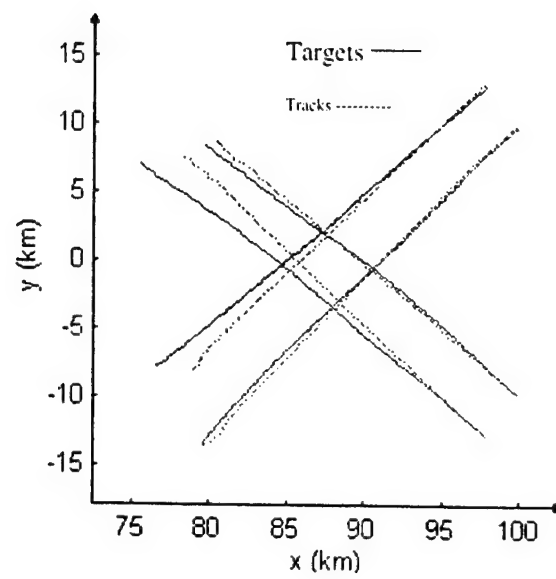


Figure 5. IR Tracks

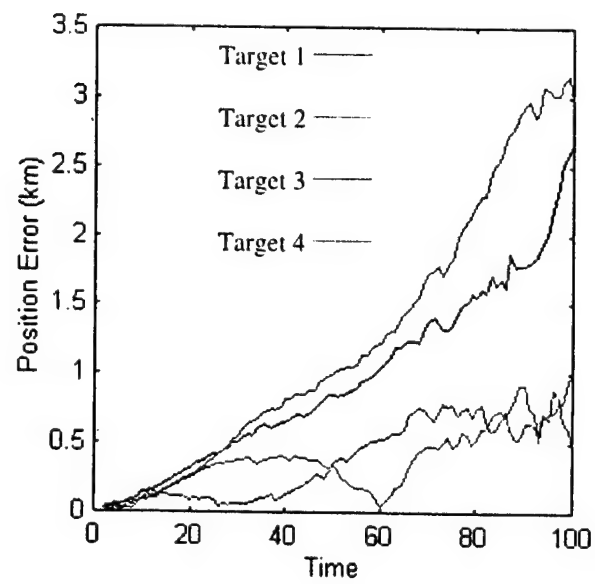


Figure 6. IR Tracker Position Errors

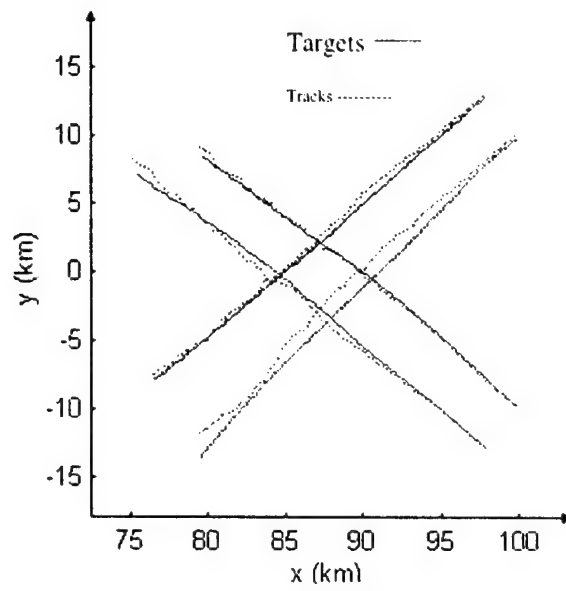


Figure 7. Radar Tracks

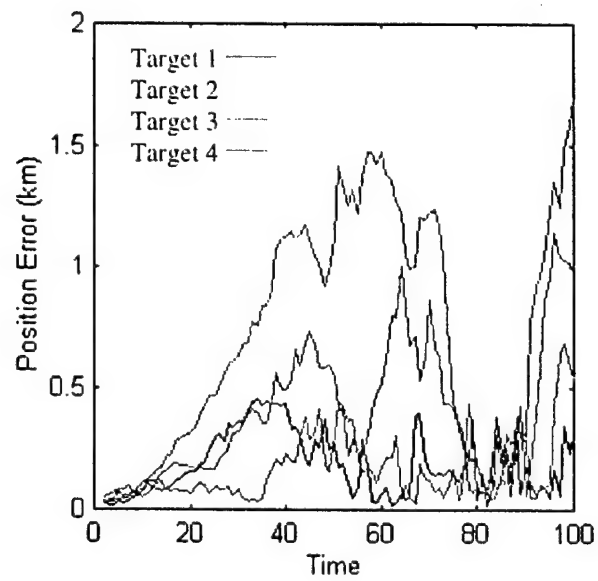


Figure 8. Radar Tracker Position Errors

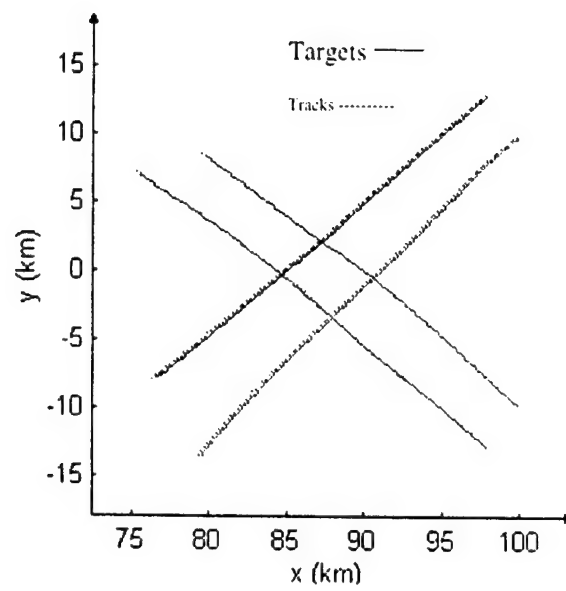


Figure 9. Laser Radar Tracks

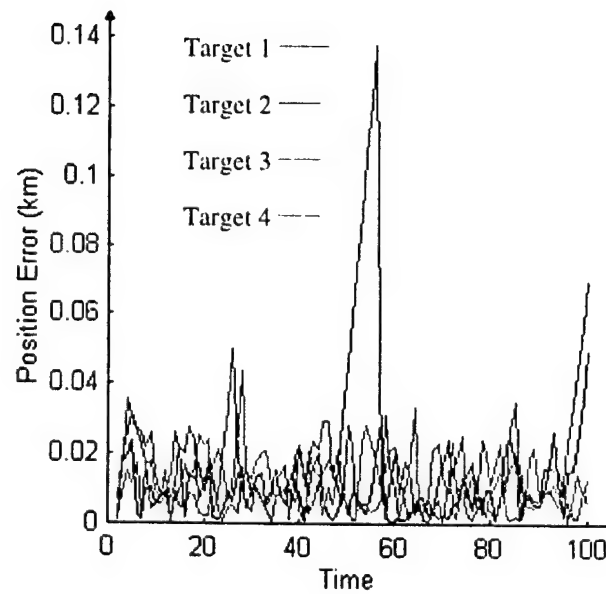


Figure 10. Laser Radar Tracker Position Errors

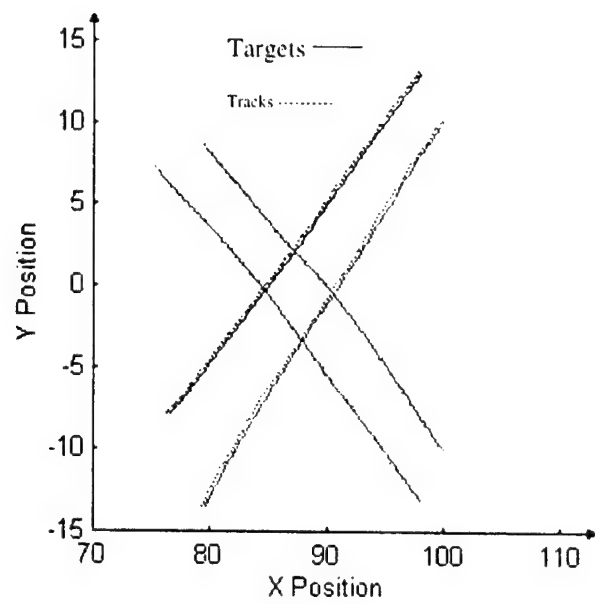


Figure 11. Fused Tracks for IR and Radar

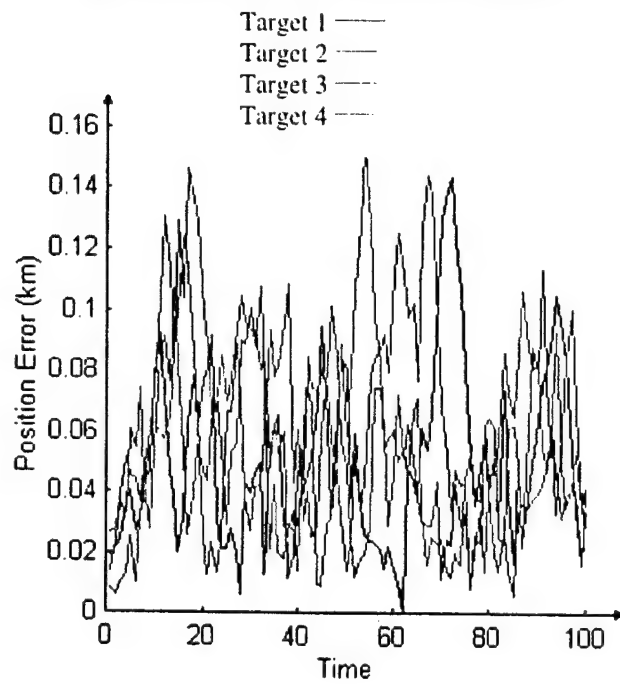


Figure 12. Fused Position Errors for IR and Radar

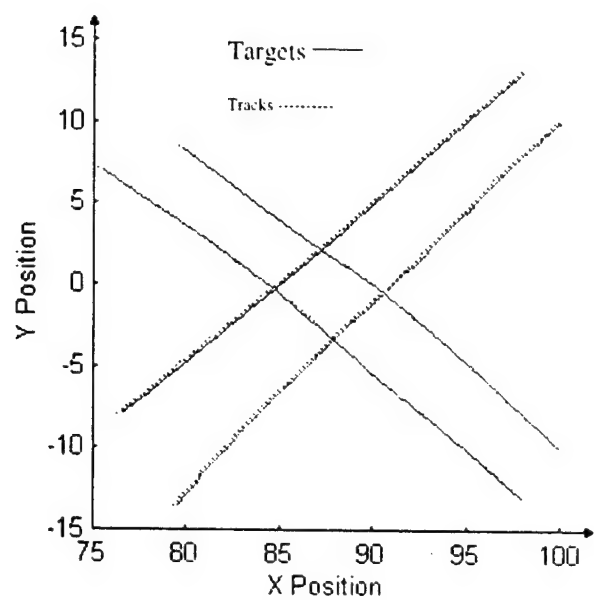


Figure 13. Fused Tracks for IR and Laser Radar

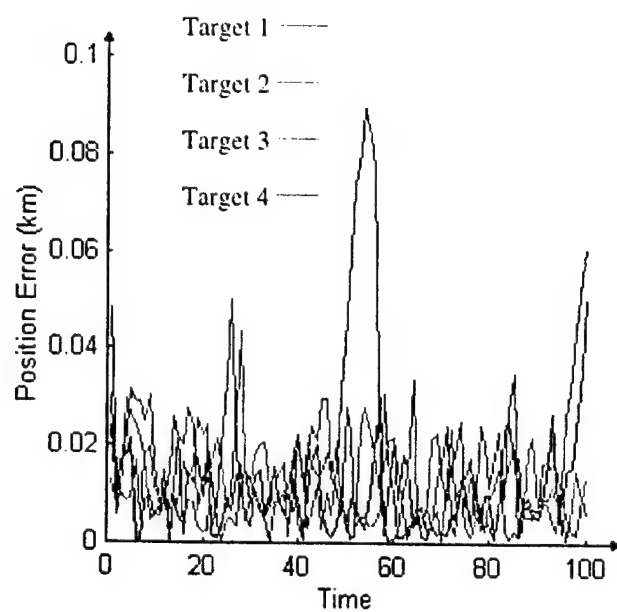


Figure 14. Fused Position Errors for IR and Laser Radar

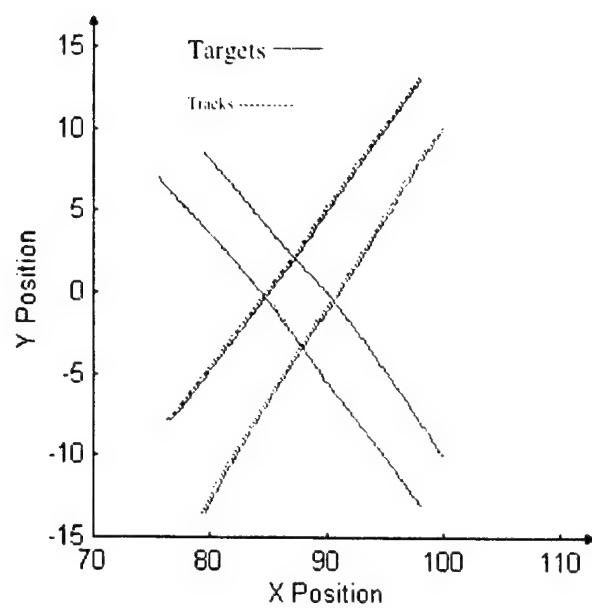


Figure 15. Fused Tracks for Radar and Laser Radar

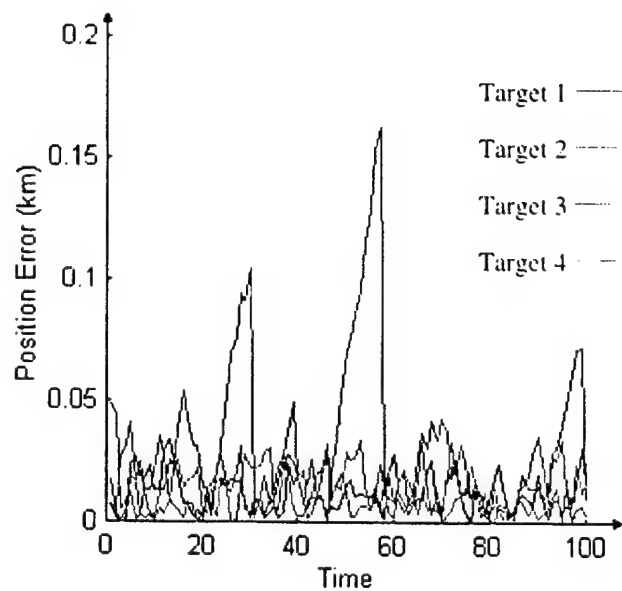


Figure 16. Fused Position Errors for Radar and Laser Radar

## 6.2 SIMULATION OF INFORMATION FUSION ALGORITHM

Track fusion algorithm employing information fusion methodology was implemented in MATLAB 4.0 under Windows 3.1 environment. This software can be installed by creating a directory, for example: c:\matlab\fusedat\nova, and by copying all the files in the diskette into the directory. To start, double click on the MATLAB icon and the command window appears. Type "cd c:\matlab\fusedat\nova" and the program can be executed interactively.

Three routines were developed to perform Monte Carlo simulation of two-sensor track fusion. Names of these routines are: demo7, demo8, and demo9. These routines can be executed by typing, say "demo7". To use the default parameters in these demos, hit <CR> on every pause/prompt for input. The default parameters are listed in the parenthesis after each prompt for input. When multiple numbers are displayed in a single request, they are the only allowable inputs and the last number is the default when no input is given. Details of the three demos are given below:

- 1) demo7 implements fusion of tracks from IR and Laser Radar sensors. This demo illustrates the results of fusion of a 4-D sensor (Laser Radar) and a 2-D sensor (IR). A centralized fusion architecture is used where the tracks from the two sensors are processed and fused at a central fusion center.
- 2) demo8 implements fusion of tracks from IR and Radar sensors. This demo illustrates the results of fusion of a 2-D sensor (IR) and a 3-D sensor (Radar). A centralized fusion architecture is used where the tracks from the two sensors are processed and fused at a central fusion center.
- 3) demo9 implements fusion of tracks from Radar and Laser Radar sensors. This demo illustrates the results of fusion of a 4-D sensor (Laser Radar) and a 3-D sensor (Radar). A centralized fusion architecture is used where the tracks from the two sensors are processed and fused at a central fusion center.



### 6.3 PERFORMANCE COMPARISON

To compare the performance of the two fusion algorithms, the example used in [3] is used as a basis for comparison. In this example, a single target is tracked by two sensors. The kinematic model of the target is given by

$$X(t_{k+1}) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} X(t_k) + \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} W(t_k)$$

with sampling rate  $\Delta T = 1$  sec. The input noise  $W(t_k)$  is assumed to be zero mean white Gaussian whose variance is  $q$ .

Observations from the two sensors, denoted by the superscript  $m$ , are modeled as

$$Z^m(t_k) = [1 \ 0]X(t_k) + v^m(t_k) \quad \text{for } m = 1, 2$$

where, sensor observation noises are assumed to be white Gaussian and independent of each other. For the sake of simplicity, it is also assumed that the sensors are similar so that  $\text{Var}[v^1(t_k)] = \text{Var}[v^2(t_k)] = 1$ . To analyze the performance of the two algorithms, the fused covariance equation (5-20) can be written as

$$P^F(t_k / t_k)^{-1} = 2P^S(t_k / t_k)^{-1} - 2P^S(t_k / t_{k-n})^{-1} + P^F(t_k / t_{k-n})$$

where, for identical sensors  $P^S(t_k / t_k) = P^1(t_k / t_k) = P^2(t_k / t_k)$  is the track variance, and

are the extrapolated covariances of the individual trackers and fused tracks respectively, and  $n$  is the number of sampling intervals between communications (fusion). The steady state fused covariance can be obtained by letting  $P^S(t_k / t_k) = P^S(t_{k-n} / t_{k-n}) = P^*$  and  $P^F(t_k / t_k) = P^F(t_{k-n} / t_{k-n}) = P$  and solving for components of  $P$ .

Figure 17. shows the ratios of the elements of the covariance matrix  $P$  and  $P^*$  for a wide range of values of the variance of the process noise  $q$ , where  $P = \begin{bmatrix} p1 & p2 \\ p2 & p3 \end{bmatrix}$ . Figure 18. shows the ratios of the areas of the ellipses of uncertainty, which are proportional to the square root of the covariance matrices. The dotted lines are the results based on (2-10), and the solid lines are the results based on (5-20) for several fusion intervals denoted by  $n$ . These figures demonstrate that the information fusion algorithm (5-20) is sensitive to the level of process noise (particularly for  $p3$ ) and the length of communication interval. This is understandable, because when the input noise variance is increased or when the time interval between fusion is longer, the impact of the underlying process noise will be significant. Figure 17. and Figure 18. also show that when the process noise variance or the communication interval is very large, ratios of elements of covariance matrices approach 0.5. This is equivalent to the case where the cross-correlation matrix  $P^C = 0$ .

It should also be noted that, even though the weighted covariance algorithm (2-9)-(2-10) is inexact, the error is small and is fairly insensitive to the level of  $q$ . This approach to track fusion, however, requires sensors from the remote stations to transmit at every  $t_k$ , the observation matrix  $H(t_k)$  and the Kalman filter gain matrix  $K(t_k)$  in addition to the track and its variance. Since the existing communication links have limited bandwidth, it may be useful to

perform a trade-off study involving the improvement due to fusion and the link bandwidth for the two approaches. Such a trade-off study could also consider the possibility of transmitting raw measurements, instead of the processed tracks, to a fusion center. Fusion of raw measurements from the remote stations leads to optimal track-to-track association and fusion [11].

When  $n=1$  (full-rate communication), the information fusion algorithm (5-20) is optimal and the results are identical to the one obtained by "measurement fusion" [15]. Figure 17. and Figure 18. provide analytical performance trade-offs for designing a fusion system under various operating conditions. For example, when the process noise is relatively small, the information fusion algorithm is a good choice because it provides almost optimal performance with minimum communication requirements. On the other hand, when the process noise is large, the weighted covariance fusion approach (2-9)-(2-10) is preferable because it incurs minor performance degradation.

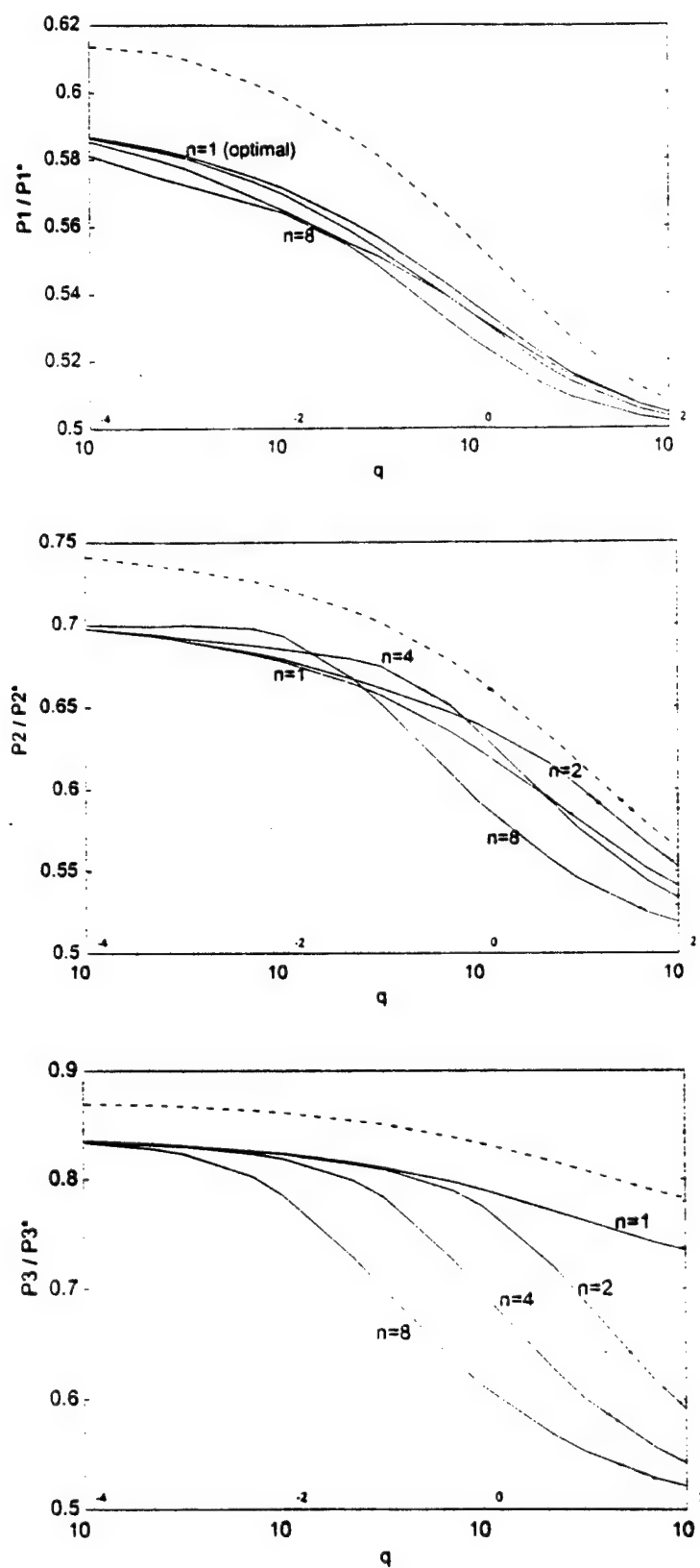


Figure 17. Ratios of Elements of Covariance Matrix for Fused State Estimates to Single Sensor State Estimates

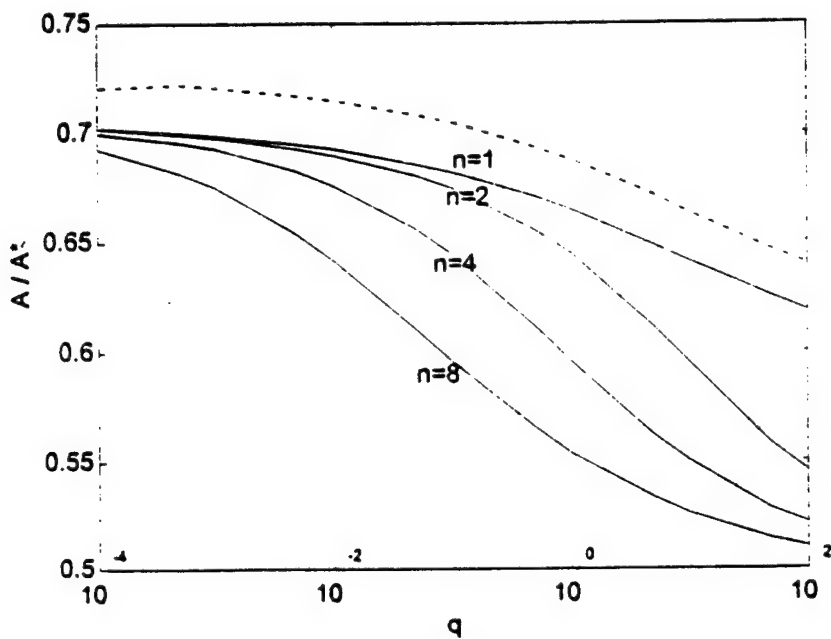


Figure 18. Ratios of Area of Error Ellipses for Fused State Estimates to Single Sensor State Estimates

## SECTION 7

### SUMMARY

This report describes the development of improved track to track fusion algorithms for C<sup>3</sup>I employing multiple dissimilar sensors. Three sensor models (Radar, IR, and Laser Radar) were generated using Monte Carlo simulation. These sensors detect four targets, perform report to track correlation and create tracks at different data rates. For this report it is assumed that the Radar, IR, and Laser Radar sensors create tracks at rates 4 sec., 2 sec., and 1 sec., respectively. These tracks are synchronized by using the KTS algorithm. The synchronized tracks are associated by employing the KTA algorithm. This algorithm associates track pairs by computing minimum closeness score. As a measure of performance, probability distributions of correct track association and false track correlation are computed by incorporating the cross-covariance in the test statistic for association. A trade-off study involving the size of the association gate, probability of correct association and probability of false correlation was also performed. Associated tracks are then fused using the KTF algorithm. This algorithm weights the tracks by certain covariances which are functions of track covariances as well as their cross-covariance, and then summed. Numerical simulation shows that the position errors of the individual trackers can be significantly reduced by performing kinematic track fusion which results in more efficient and improved operation of airborne surveillance systems.

In addition, a KTF algorithm based on information fusion approach was also developed. This algorithm is efficient because its implementation requires less computation but does not incorporate the cross-covariance between the candidate tracks for fusion. Performances of the two algorithms are compared by means of covariance analysis.

## **LIST OF REFERENCES**

1. R. Singer and A.J. Kanyuck, "Computer Control of Multiple Site Track Correlation," *Automatica*, Vol. 7, pp 455-464, July 1971.
2. Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," *IEEE Trans. on Automatic Control*, Vol. AC-26, No. 2, pp 571-572, April 1981.
3. Y. Bar-Shalom, and L. Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused Track Covariance," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. AES-22, No. 6, pp 803-805, November 1986.
4. R. Saha, "Effect of Common Process Noise on Two-Sensor Track Fusion," to be published in the *AIAA Journal of Guidance, Control and Dynamics*.
5. R. Saha, "Track-to-Track Fusion with Dissimilar Sensors," to be published in the *IEEE Trans. on Aerospace and Electronic Systems*.
6. K. Chang and R. Saha, "On Optimal Track-to-Track Fusion," to be published in the *IEEE Trans. on Aerospace and Electronic Systems*.
7. Y. Bar-Shalom and T.E. Fortman, *Tracking and Data Association*, Academic Press, CA, 1988.
8. Y. Bar-Shalom and X. Rong Li, "Effectiveness of the Likelihood Function in Logic-Based Track Formation," *IEEE Trans. on Aerospace and Electronic Systems*, AES-27, No. 1, pp 184-187, January 1991.

9. C.Y. Chong, S. Mori, and K.C. Chang, "Distributed Multitarget Multisensor Tracking," in Multitarget-Multisensor Tracking: Applications and Advances, Vol. I, Chapter 8, Edited by Y. Bar-Shalom, Artech House, 1990.
10. R. Lobbia and M. Kent, "Data Fusion of Decentralized Local Tracker Outputs," IEEE Trans. on Aerospace and Electronic Systems, AES-30, No. 3, pp 787-798, July 1994.
11. S. Mori, K. Demetri, W. Barker, and R. Lineback, "A Theoretical Foundation of Data Fusion," Proc. Data Fusion Symposium, pp 585-594, 1994.
12. S. Hovanessian, Radar System Design and Analysis, Artech House, MA, 1984.
13. K. Seyrafi and S. Hovanessian, Introduction to Electro-Optical Tracking System, Artech House, 1992.
14. A. V. Jelalian, Laser Radar Systems, Artech House, 1992.
15. J.A. Roecker and C.D. McGillem, "Comparison of Two-Sensor Tracking Methods Based on State Vector Fusion and Measurement Fusion," IEEE Trans. on Aerospace and Electronic Systems, AES-24, No. 4, pp 447-449, July 1988.
16. B. Friedland, "Optimum Steady-State Position and Velocity Estimation Using Noisy Sampled Position Data," IEEE Trans. on Aerospace and Electronic Systems, AES-9, No. 6, pp 906-911, November 1973.
17. A. S. Willsky, M.G. Bello, D.A. Castanon, B.C. Levy, and G.C. Verghese, "Combining and Updating of Local Estimates and Regional Maps Along Set of One-Dimensional Tracks," IEEE Trans. on Automatic Control, AC-27, No. 4, pp 799-813, August 1982.



## APPENDIX A

### DESCRIPTION OF SENSOR MODELS

This section describes the parameters of the three sensors (Radar, IR, and Laser Radar) which are used to detect the targets. In this report, these sensors are assumed to be colocated. After detection, the sensors perform report to track correlation. These tracks are transmitted to a central fusion center where track fusion is performed.

#### Radar Sensor Model

The radar model used in this simulation is described in [12]. Output of the radar is the observation vector [Range, Azimuth, Elevation]. The equation describing the range and signal to noise ratio is given by

$$R = \frac{\bar{P} A \sigma L}{16(kT)(S/N)} (t_r / \Omega)^{1/4}$$

Nominal values of the critical parameters are used in this study are given below:

$$\bar{P} = 500 \text{ Watts}$$

$$D \text{ (diameter of aperture)} = 1 \text{ ft}$$

$$A = \pi(1)^2 / 4(3.28)^2] = 0.073 \text{ m}^2 \text{ aperture area}$$

$$k = 1.38 \times 10^{-23} \text{ Joule/}^{\circ}\text{K} = \text{Boltzman constant, } T = 600^{\circ}\text{K}$$

$$\sigma = 10 \text{ m}^2 \text{ (small aircraft)}$$

$$L = L_{\text{radar}} \times L_{\text{atmos}}, L_{\text{radar}} = \text{Radar (transmit/receive) Losses} = 0.1$$

$$L_{\text{atmos}} = \text{Atmospheric Losses (from 3GHz to 8GHz)} = 0.6$$

$$L = 0.1 \times 0.6 = 0.06$$

$$\Omega = \text{Angular scan coverage} = 100 \times (120^{\circ}) \times (\pi/180)^2 = 0.365 \text{ steradians}$$

$t_s$  = Scan duration = 3 sec.

$\lambda$ (at 8 GHz) =  $3 \times 10^8 / 8 \times 10^9 = 0.0375$  m

$\theta_{BW} = 0.0375 / 0.305 = 0.123$  radians =  $7^\circ$  (elevation)

Assuming a circular aperture, null-null beamwidth =  $2 \times 7^\circ = 14^\circ$

$\theta_{BW}$  (azimuth) =  $2^\circ$

### IR Sensor model:

The IR sensor model used in this simulation is described in [13]. Output of the IR sensor is the observation vector (Azimuth, Elevation). The equation describing the range and signal to noise ratio is given by

$$R = \left[ \frac{\pi D J D^* L}{4(f\#)(S/N)} \right]^{1/2} \left( \frac{nt_s}{\Omega} \right)^{1/4}$$

Nominal values of the critical parameters are used in this study are given below

$D$  = Optical aperture = 1 ft. = 30.48 cm.

$D^*$  = Dielectric constant =  $2 \times 10^{10}$  cm-HZ<sup>1/2</sup>/Watts

$L$  = System and atmospheric losses = 0.1,

$n$  = Number of resolution elements(detectors)= 3600 (600 in elevation and 6 in azimuth)

$f\#$  = Focal number = 1.5,  $t_s$  = Frame time = 2 sec.

$\Omega$  = Total solid angle coverage =  $120^\circ(\text{ele}) \times 10^\circ(\text{azi}) \times (\pi/180)^2 = 0.365$  steradians

$J$  = Source radiation intensity = 40 watts/ steradians

$\theta_{BW}$  = Beamwidth =  $\lambda/D = 10^{-5}$  m/ 0.3048 m = 32.8 m rad = 0.002 degrees

Resolution in elevation = angular coverage/ number of detectors =  $10^\circ/600 = 0.0166$  deg

= 0.29 m rad

### Laser Radar Sensor Model

The Laser Radar sensor model used in this simulation is described in [14]. Output of the Laser Radar sensor is the observation vector (Range, Range Rate, Azimuth, Elevation). Sensor model given below is based on the assumption of a CO<sub>2</sub> laser. For a coherent receiver, the equation describing the range and signal to noise ratio (for a point target) is given by

$$R = \left(\frac{D}{2}\right) \left(E_T \frac{\sigma \eta_{SYS} \eta_{ATM}}{\lambda (S/N) hc}\right)^{1/4}$$

Nominal values of the critical parameters are used in this study are given below

$R$  = system range to target =  $10^5$  meter

$E_T$  = transmitted laser energy = 1

$S/N$  = signal to noise ratio = 31.4

$D$  = receiver aperture diameter = 0.1 m,

$\eta_{ATM}$  = atmospheric transmission factor = 0.1

$\eta_{SYS}$  = system transmission factor = 0.1

$N$  = Quantum noise of coherent optical receiver

$h$  = Planck's constant =  $6.626 \times 10^{-34}$  J-s

$\lambda$  = Wavelength of radiation =  $10 \mu\text{m}$

Standard deviation of the measurement error in Range is =  $\frac{c\tau}{2\sqrt{2(S/N)}}$

Standard deviation of the measurement error in Doppler is =  $\frac{\sqrt{3}\lambda}{\pi\tau\sqrt{2(S/N)}}$

Standard deviation of Azimuth and Elevation is =  $\frac{3\pi\tau}{16D\sqrt{(S/N)}}$

where,  $\tau$  = pulse width. For example, the nominal system parameters of  $S/N = 50$ ,  $\tau = 10^{-6}$  sec.,  $\lambda = 10^{-5}$  meter, and  $D = 0.1$  meter would give rise to the following measurement noise standard deviations:  $\sigma_R \approx 15$  m,  $\sigma_D \approx 0.5$  m,  $\sigma_A = \sigma_E = 10^{-5}$  rad. For the targets within the sensor field of view, the detection model [14] can be described as

$$P_D = P_{FA}^{\frac{1}{1+(S/N)}}$$

where,  $P_{FA}$  is the probability of false alarm.

***MISSION***  
***OF***  
***ROME LABORATORY***

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

- a. Conducts vigorous research, development and test programs in all applicable technologies;
- b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;
- c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;
- d. Promotes transfer of technology to the private sector;
- e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.